

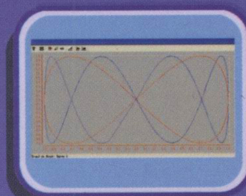
自动化技术轻松入门丛书

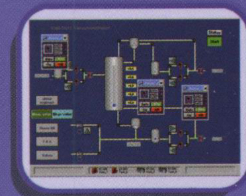
# WinCC组态技巧与技术问答

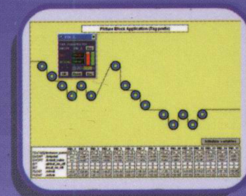

李　军◎编著














DVD ROM

- 软件：WinCC V6.2+STEP 7 V5.4
- WinCC中文手册和技术资料
- WinCC演示项目






机械工业出版社
CHINA MACHINE PRESS

VISIT…

LANZAROTE Caliente.com

自动化技术轻松入门丛书

# WinCC 组态技巧与技术问答

李　军　编著



机 械 工 业 出 版 社

本书按照组态HMI系统的一般过程，系统地介绍了组态软件WinCC在创建HMI项目、组态HMI系统与控制器的通信连接、组态过程监控画面、组态消息系统、组态过程变量记录系统、组态数据报表系统等过程中涉及的基本概念、基本理论以及组态方法和组态技巧，对用户在应用WinCC时遇到的一些技术问题和难题进行了详细的分析并提出了有效的解决方法。

本书可作为高等院校机械工程及自动化、电气工程及自动化、计算机控制等相关专业的教材，也可作为企业现场工程技术人员的培训教材或参考书。

**图书在版编目（CIP）数据**

WinCC组态技巧与技术问答/李军，编著.—北京：机械工业出版社，2013.1(2015.5重印)
（自动化技术轻松入门丛书）
ISBN 978-7-111-40929-8

Ⅰ.①W…　Ⅱ.①李…　Ⅲ.①可编程序控制器　Ⅳ.①TM571.6

中国版本图书馆CIP数据核字（2012）第302940号

机械工业出版社（北京市百万庄大街22号　邮政编码100037）
责任编辑：时　静
责任印制：李　洋
三河市宏达印刷有限公司印刷
2015年5月第1版·第2次印刷
184mm×260mm·21.25印张·527千字
3501–5000册
标准书号：ISBN 978-7-111-40929-8
　　　　　ISBN 978-7-89433-761-0（光盘）
定价：59.00元（含1DVD）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

| 电话服务 | 网络服务 |
| --- | --- |
| 社服务中心：(010)88361066 | 教材网：http://www.cmpedu.com |
| 销售一部：(010)68326294 | 机工官网：http://www.cmpbook.com |
| 销售二部：(010)88379649 | 机工官博：http://weibo.com/cmp1952 |
| 读者购书热线：(010)88379203 | **封面无防伪标均为盗版** |

# 前　言

随着计算机和网络通信技术的发展，现代工业自动化生产过程一般采用 4 级控制系统，即 0 级交直流传动、1 级基础自动化、2 级工程自动化和 3 级生产制造控制（MES）。其中，监控与数据采集系统（Human Machine Interface/Supervisory Control and Data Acquisition，HMI/SCADA）是整个控制系统的重要组成部分，实现了控制过程的可视化，为现场人员监控自动化生产过程提供了良好的用户平台。

德国西门子（Siemens）公司是全球自动化系统的领导产商，其自动化系统的核心技术——PLC 技术和总线技术都处于世界的最前沿。WinCC 是西门子全集成自动化（TIA）产品家族的重要成员，与西门子的自动化系统可以无缝集成，在全球及国内拥有极高的市场占有率。例如，在国内的钢铁、化工、能源、酿酒、烟草、造纸等行业拥有众多的客户。而随着西门子自动化产品和 WinCC 在国内各行各业的逐步推广，对掌握西门子自动化系统技术和组态软件技术的人才的需求也在不断地增长。顺应这种形势，国内很多高校也在相关专业专门开设了西门子自动化系统技术和组态软件技术的课程。

为了使读者能够迅速地熟悉 WinCC 的基本功能，以及熟练地掌握这些功能的应用范围、应用方法和组态技巧，本书从以下几个方面着手，在编写形式、结构体系、内容以及立意等方面进行了创新，使得本书有着比较鲜明的特色。

（1）本书按照组态 HMI 系统的基本过程，详细阐述了创建 HMI 项目、组态 HMI 系统与控制器的通信连接、组态过程监控画面、组态消息系统、组态过程变量记录系统、组态数据报表系统等涉及的基本概念和基本理论。

（2）本书舍弃了众多软件类书籍软件功能说明书的模式，通过大量的应用实例，详细介绍了 WinCC 组态 HMI 项目时应用的一些方法和技巧，让用户在掌握组态方法和组态技巧的同时，潜移默化地熟悉 WinCC 软件的功能，理解和掌握组态软件的基本概念和基本理论。

（3）本书为了满足广大 WinCC 用户的实际需要，解决了一些 WinCC 常见的技术问题和难题。这些技术问题和难题有些是编者多年使用 WinCC 过程中曾经遇到过的，有的是编者在培训企业现场人员过程中由学员提出的，还有的是用户通过网络技术论坛提出的，多数用户在遇到这些问题和难题时都很难下手解决，本书对这些技术问题和难题的表现形式、出现原因以及有效解决方案都进行了详细的阐述。

为了方便读者上机操作，本书配有信息丰富的 DVD 光盘，内容包括 WinCC V6.2 ASIA 软件、WinCC 技术手册和技术资料以及示例程序。

本书由北京联合大学机电学院李军编写，在写作过程中得到了北京联合大学机电学院陈瑞阳教授和刘长青老师的大力支持与指导，在此表示衷心感谢！

由于编者水平有限，错误和疏漏之处在所难免，恳请批评指正，并提出宝贵意见，对此深表谢意。编者的 E-mail 地址：jdtlijun@buu.edu.cn

编　者



# 目　录

前言
第 1 章　WinCC 系统概述 …… 1
1. 1　WinCC 简介 …… 1
1. 2　WinCC 的基本功能 …… 1
1. 3　WinCC 的特点 …… 3
1. 4　WinCC 的体系结构 …… 3
1. 4. 1　WinCC 基本系统 …… 4
1. 4. 2　WinCC 选件 …… 5
1. 4. 3　WinCC 附加件 …… 6
第 2 章　WinCC 的安装 …… 8
2. 1　WinCC 的安装要求 …… 8
2. 1. 1　安装 WinCC 的硬件要求 …… 8
2. 1. 2　安装 WinCC 的软件要求 …… 9
2. 2　WinCC 安装前的兼容性检查 …… 10
2. 3　WinCC V6. 2 的安装 …… 10
2. 3. 1　安装注意事项 …… 10
2. 3. 2　安装步骤 …… 11
2. 4　WinCC 的补充安装和卸载 …… 23
2. 5　WinCC V6. 2 安装的常见问题及解决办法 …… 24
2. 5. 1　Windows XP 无法安装消息队列 …… 24
2. 5. 2　安装时提示无法找到 ssf 文件 …… 24
2. 5. 3　安装时反复要求重新启动计算机 …… 25
2. 5. 4　安装 WinCC 后 Windows 系统无法正常登录 …… 26
2. 5. 5　Windows XP SP3 安装 WinCC V6. 2 时无法安装 KB319740 …… 27
第 3 章　WinCC 的项目管理和变量管理 …… 29
3. 1　WinCC 的项目管理 …… 29
3. 1. 1　WinCC 的系统架构 …… 29
3. 1. 2　WinCC 项目的类型 …… 31
3. 1. 3　WinCC 项目的创建 …… 32
3. 1. 4　WinCC 项目的文件结构 …… 36
3. 1. 5　WinCC 项目的激活 …… 37
3. 2　WinCC 的变量管理 …… 38
3. 2. 1　WinCC 变量的功能类型 …… 38
3. 2. 2　WinCC 变量的数据类型 …… 39




3.2.3　WinCC 的通信 …… 41
3.2.4　WinCC 变量的创建 …… 45
3.3　组态技巧与技术问答 …… 50
3.3.1　移植（升级）WinCC 项目的方法 …… 50
3.3.2　复制 WinCC 项目的方法 …… 52
3.3.3　WinCC 项目无法正常激活的解决办法 …… 52
3.3.4　WinCC 出现“死机”现象的解决办法 …… 53
3.3.5　WinCC 项目中直接调用 STEP 7 变量的方法 …… 54
3.3.6　WinCC 变量的导入/导出方法 …… 60
3.3.7　诊断 WinCC 变量状态的方法 …… 63
第 4 章　过程画面的组态及用户权限管理 …… 64
4.1　WinCC 图形编辑器 …… 64
4.1.1　图形编辑器的组成 …… 65
4.1.2　图形编辑器的基本操作 …… 69
4.1.3　图形编辑器的库 …… 72
4.2　动态变化的触发器和更新周期 …… 73
4.3　组态过程画面 …… 74
4.3.1　设计过程画面的结构 …… 74
4.3.2　设计过程画面的布局 …… 74
4.3.3　创建过程画面 …… 75
4.3.4　过程画面的动态化 …… 77
4.4　组态动态化举例 …… 79
4.4.1　组态对话框 …… 79
4.4.2　动态向导 …… 81
4.4.3　变量连接 …… 84
4.4.4　动态对话框 …… 86
4.4.5　直接连接 …… 92
4.4.6　C 动作 …… 96
4.4.7　VBS 动作 …… 101
4.5　画面模板 …… 105
4.5.1　画面模板的概念 …… 105
4.5.2　变量前缀的概念 …… 106
4.5.3　变量前缀方式应用画面模板的原理 …… 106
4.5.4　变量前缀方式组态实例 …… 107
4.6　WinCC 用户权限管理 …… 113
4.6.1　WinCC 用户管理器 …… 114
4.6.2　组态用户管理 …… 116
4.6.3　组态画面对象的访问保护 …… 117
4.6.4　组态用户登录和注销的对话框 …… 118



4.6.5 使用与用户登录相关的内部变量 …… 120
4.7 组态技巧与技术问答 …… 120
4.7.1 应用 S7 – PLCSIM 和 WinCC 变量模拟器测试 WinCC 项目 …… 120
4.7.2 WinCC 过程画面显示系统信息的组态方法 …… 124
4.7.3 禁用 Windows 组合键的组态方法 …… 126
4.7.4 在 Windows 组合键被禁用时从 WinCC 运行系统进入组态系统 …… 126
4.7.5 二元开关操作的组态方法 …… 127
4.7.6 颜色更改动态的组态方法 …… 131
4.7.7 移动过程动画的组态方法 …… 137
4.7.8 复选框应用于位操作的组态方法 …… 141
第 5 章 消息系统 …… 143
5.1 WinCC 消息系统的相关概念 …… 143
5.1.1 消息和报警 …… 143
5.1.2 确认 …… 146
5.1.3 归档 …… 146
5.2 消息系统的组态 …… 148
5.2.1 报警记录编辑器 …… 148
5.2.2 组态消息系统的基本步骤 …… 149
5.2.3 启动报警记录的系统向导 …… 149
5.2.4 组态消息块 …… 150
5.2.5 组态消息类别和消息类型 …… 152
5.2.6 组态单个消息 …… 155
5.2.7 组态模拟量报警 …… 159
5.2.8 组态消息归档 …… 161
5.3 在运行系统中显示消息 …… 165
5.3.1 组态 WinCC 报警控件 …… 165
5.3.2 报警控件在运行系统中的操作 …… 166
5.3.3 在运行系统中显示归档的消息 …… 168
5.4 应用实例 …… 169
5.4.1 组态要求 …… 169
5.4.2 组态方法与步骤 …… 171
5.5 组态技巧与技术问答 …… 175
5.5.1 消息窗口无法显示消息的解决方法 …… 175
5.5.2 WinCC 实现语音报警的组态方法 …… 176
5.5.3 WinCC 组态来自于 PLC 的编号消息的方法 …… 177
5.5.4 组态在报警消息中显示系统当前登录用户的方法 …… 182
5.5.5 通过 C 脚本确认报警消息的组态方法 …… 184
第 6 章 变量记录系统 …… 186
6.1 WinCC 变量记录系统的相关概念 …… 186

6.1.1　过程值归档的原理 …… 186
6.1.2　归档方法 …… 188
6.1.3　过程值的存储 …… 189
6.2　组态过程值归档 …… 190
6.2.1　组态变量记录系统的步骤 …… 191
6.2.2　WinCC 变量记录 …… 191
6.2.3　组态定时器 …… 191
6.2.4　在变量记录中创建归档 …… 192
6.2.5　创建并组态归档变量 …… 193
6.2.6　组态归档 …… 199
6.2.7　组态归档备份 …… 200
6.3　过程值归档的输出 …… 201
6.3.1　以趋势的形式输出过程值 …… 202
6.3.2　以表格形式输出过程值 …… 210
6.4　应用实例 …… 214
6.4.1　组态要求 …… 214
6.4.2　组态方法与步骤 …… 214
6.5　组态技巧与技术问答 …… 219
6.5.1　WinCC 如何界定快速归档与慢速归档? …… 219
6.5.2　归档数据库尺寸的计算方法 …… 220
6.5.3　合理设置过程值归档的采集周期和归档周期的方法 …… 220
6.5.4　整点启动过程值归档的组态方法 …… 221
第 7 章　报表系统 …… 225
7.1　WinCC 报表系统简介 …… 225
7.2　WinCC 报表编辑器 …… 226
7.2.1　页面布局 …… 226
7.2.2　行布局 …… 230
7.2.3　打印作业 …… 232
7.3　组态报表 …… 236
7.3.1　组态行式打印机上的消息顺序报表 …… 237
7.3.2　组态页面布局的消息顺序报表 …… 238
7.3.3　组态归档消息报表 …… 241
7.3.4　组态变量记录报表 …… 242
7.4　组态技巧与技术问答 …… 244
7.4.1　项目文档的输出方法 …… 244
7.4.2　输出硬拷贝（画面拷屏）的方法 …… 245
7.4.3　组态在过程画面中启动报表输出的按钮的方法 …… 247
7.4.4　将 WinCC 归档的历史数据导出到 Excel CSV 文件的方法 …… 248
7.4.5　WinCC 组态复杂报表的方法 …… 261



第 8 章　全局脚本及开放性接口 …… 267
8.1　全局脚本 …… 267
8.1.1　ANSI-C 脚本 …… 267
8.1.2　VBScript …… 276
8.1.3　应用实例 …… 284
8.2　WinCC 的开放性接口 …… 289
8.2.1　WinCC 的开放性接口概貌 …… 289
8.2.2　OLE 接口及其应用 …… 291
8.2.3　DDE 接口及应用 …… 294
8.2.4　API 接口及应用 …… 297
8.2.5　使用 ActiveX 控件 …… 299
8.2.6　ADO/OLE－DB 访问归档数据库 …… 300
8.3　组态技巧与技术问答 …… 306
8.3.1　在 WinCC 中读取系统时间的方法 …… 306
8.3.2　用 C 脚本解决授权点数不够用问题的方法 …… 307
8.3.3　用 C 脚本实现用户登录日志 …… 308
第 9 章　基于 OPC 的数据访问 …… 310
9.1　OPC 的基本概念 …… 310
9.1.1　OPC 的优点 …… 310
9.1.2　OPC 基金会 …… 311
9.1.3　OPC 的层次对象模型 …… 311
9.1.4　OPC 的数据访问方式 …… 312
9.1.5　OPC 服务器的自定义和自动化接口 …… 313
9.2　OPC 在 WinCC 中的应用 …… 313
9.2.1　WinCC 支持的 OPC 规范 …… 313
9.2.2　在 WinCC 中使用 OPC …… 314
9.3　WinCC 基于 OPC 的数据访问应用实例 …… 316
9.3.1　WinCC 到 WinCC 的连接 …… 316
9.3.2　WinCC 到 Microsoft Excel 的连接 …… 318
9.3.3　WinCC 与 SIMATIC NET 的 OPC DA 连接 …… 323
9.3.4　WinCC 与 S7－200 PLC 的 OPC DA 连接 …… 326
附录 …… 330
附录 A　C 运算符 …… 330
附录 B　VB 运算符 …… 331
参考文献 …… 332

# 第1章

# WinCC 系统概述

本章介绍 WinCC 的有关知识，主要包括 WinCC 简介、WinCC 的基本功能、WinCC 的特点和 WinCC 的体系结构等。通过本章内容的介绍，读者可以对 WinCC 有初步了解，为学习本书后续内容作必要的准备。

## 1.1 WinCC 简介

WinCC（Windows Control Center，视窗控制中心）是由西门子公司推出的基于 Windows 操作系统的功能强大的组态软件。组态软件是 HMI/SCADA 系统的软件平台工具，HMI 代表人机接口（Human Machine Interface），SCADA 代表数据采集监控系统（Supervisory Control and Data Acquisition）。HMI/SCADA 系统是人（即操作员）和机器（即过程）之间的界面，目前已广泛应用于工业、农业、楼宇等领域的自动化系统中。在这样的自动化系统中，系统控制器（如 PLC）保持对过程的实际控制，而 HMI/SCADA 系统用于实现过程的可视化，为操作员提供友好的图形用户界面（Graphics User Interface）。

WinCC 是 HMI/SCADA 系统组态软件中的后起之秀，于 1996 年进入工控组态软件市场，当年就被美国《Control Engineering》杂志评选为最佳 HMI 软件，从而以最短的时间发展成第三个在世界范围内获得成功的 HMI/SCADA 系统。

作为全集成自动化（Totally Integrated Automation，TIA）系统的重要组成部分，WinCC 确保与西门子的自动化系统——SIMATIC S5、S7 和 505 系列 PLC 进行方便的连接和高效的通信；WinCC 与 STEP 7 的紧密结合，将 WinCC 项目集成到 STEP 7 项目中，可以实现在 WinCC 里直接调用 STEP 7 的符号表或 DB 块里的变量，这种统一的数据管理机制，大大地提高了工作效率，降低了出错概率。

WinCC 的设计思想秉承了西门子公司博大精深的企业文化理念，开发者努力追求 WinCC 成为性能最全面、技术最先进、系统最开放的 HMI/SCADA 系统。WinCC 从一开始就按照世界范围内使用的控制系统进行设计，因此适合于世界上各主要制造商生产的控制系统。而通过 OPC（OLE for Process Control，用于过程控制的 OLE 接口）方式，WinCC 实现了与更多的第三方控制器进行通信。

## 1.2 WinCC 的基本功能

WinCC 是运行于 32 位 Windows 操作系统的组态软件，采用类似于 Windows 资源管理器的窗口结构，集成所有的 SCADA 功能，基本功能包括：对自动化系统中的各种资源



（如设备、过程变量、过程可视化画面等）进行配置和编辑；处理数据报警和系统报警；提供多种通信驱动程序；提供多种标准接口保证系统的开放性，实现与其他应用程序的数据共享；各类报表的创建和打印输出；提供脚本系统实现软件的二次开发功能；存储历史数据并支持历史数据的查询等。WinCC 的基本功能详细介绍如下：

- 过程可视化：WinCC 允许操作员对过程进行观察，过程以图形化的方式显示在屏幕上，过程的状态一旦发生变化，画面将动态更新；WinCC 还允许操作员通过画面控制过程，例如，操作员可以通过图形用户界面预置控件的参数或打开阀门等。
- 报警显示（消息系统）：一旦出现临界过程状态，将自动触发报警信号。例如，如果超过了预定义的限制值，画面上将显示一条报警消息。
- 过程值归档（变量记录系统）：WinCC 运行系统支持以趋势曲线或表格的形式在画面中实时显示过程值，并打印输出生产数据，也支持连续、顺序地记录过程值和报警（即归档），允许以 OPC HDA 接口等方式对历史的生产数据进行访问。
- 报表系统：WinCC 提供报表的创建和打印输出功能。报表的创建是在 WinCC 的报表编辑器中编辑报表的布局。WinCC 允许创建和输出项目文档报表和运行系统数据报表，项目文档报表即 WinCC 项目的组态数据，而运行系统数据报表可在运行期间输出过程数据。
- 过程通信：实现 WinCC 和控制器（主要是 PLC）之间的通信连接。WinCC 除了提供专用的通道，用于连接 SIMATIC S5/S7/S505 等系列的 PLC，还支持 PROFIBUS DP/FMS、DDE（Dynamic Data Exchange，动态数据交换）和 OPC 等通用通道连接第三方控制器。另外，WinCC 还提供了一个 CDK（通道开发软件包）选件，用于开发一些专用的通信通道。
- 标准接口：对标准接口的支持允许 WinCC 与其他控制器和应用程序共享数据，即开放性。实际上，WinCC 代表了最高水平的开放性和集成性，它几乎集成了 Microsoft 所有的开放性技术，包括 ActiveX、DDE、OLE、ODBC、OPC 以及 Microsoft 强大的数据库 Microsoft SQL Server。
- 脚本系统：实现了 WinCC 的二次开发功能，复杂的过程可视化任务可以通过脚本系统来完成。WinCC 脚本系统提供两种脚本语言：ANSI-C 和 VBScript。

WinCC 的基本功能示意图如图 1-1 所示。

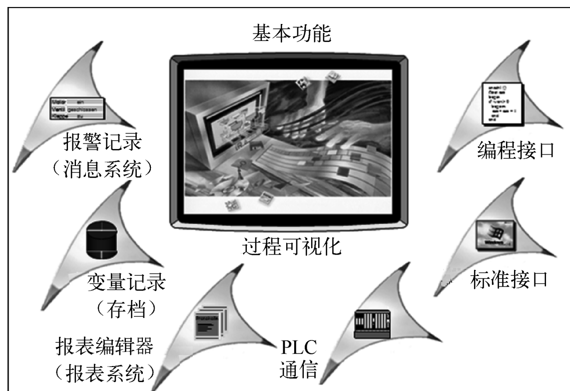



图 1-1　WinCC 基本功能示意图

## 1.3　WinCC 的特点

WinCC 能够以最短的时间发展成为第三个在世界范围内获得成功的 HMI/SCADA 系统，是与其强大的功能和鲜明的特点密不可分的。WinCC 具有以下特点：

- WinCC 是基于 Windows 操作系统的组态软件。西门子公司与微软公司的密切合作保证了 WinCC 基于最新发展的软件技术进行开发，让用户获得不断创新的技术。
- 可以优化定制 WinCC，以满足过程的实际需要。WinCC 支持大范围的组态可能性，根据系统的复杂程度，可以组态从单用户系统和客户机 - 服务器系统一直到具有多台服务器的冗余分布式系统。
- WinCC 提供众多的选件（options）和附加件（add-on）来扩展其基本功能，这些选件和附加件均基于开放式编程接口，覆盖不同行业的需求。例如，Audit（操作和跟踪）选件可以跟踪操作员的信息（如用户名、操作日期等）和操作内容，其应用可以让 WinCC 满足 FDA（美国食品和药物管理局）的需求。
- 作为 Siemens TIA 的重要部件，WinCC 可以与属于 SIMATIC 产品家族的自动化系统非常协调地进行工作，因为 TIA 意味着 SIMATIC 产品在组态、编程、数据存储和通信等方面的一致性。
- 通过标准化接口，WinCC 可与其他 IT 解决方案交换数据。例如，与 MES（Manufacturing Execution Systems，制造执行系统）和 ERP（Enterprise Resource Planning，企业资源计划）层的应用程序（如 SAP 系统）共享数据，可以将 WinCC 从过程监控扩展到工厂级监控，为企业管理 MES 和 ERP 提供管理数据。
- 开放的 WinCC API 编程接口，允许用户开发集成了 WinCC 的部分功能的程序，从而能够控制过程和过程数据。
- WinCC 提供大量向导方式的简易组态，简化了组态工作，提高了组态效率。WinCC 还支持调试阶段的在线修改。
- WinCC 是一种与 Internet 兼容的 HMI/SCADA 系统，这种系统容易实现基于 Web 的客户机解决方案以及瘦型客户机解决方案。
- WinCC 是基于多语言设计的，用户可以在英语、德语、法语以及其他众多语言之间进行选择。支持系统运行时的在线语言切换，这意味着 WinCC 可以在几个目标市场使用相同的可视化解决方案。

## 1.4　WinCC 的体系结构

WinCC 具有模块化的结构，它由 WinCC 基本系统组成，并提供了许多 WinCC 选件和 WinCC 附加件。WinCC 的集成开发环境称为 WinCC Explorer，图 1-2 为 WinCC 的体系结构示意图，该示意图明确地显示了 WinCC 的各应用模块之间的关系。从外观和操作来看，WinCC Explorer 采用了一种类似于 Windows 资源管理器（Windows Explorer）的结构，它组合了操作 WinCC 项目的所有必要的数据，这些数据都是分层排列和存储的，用户只要很少的操作就可以编辑和浏览这些数据。



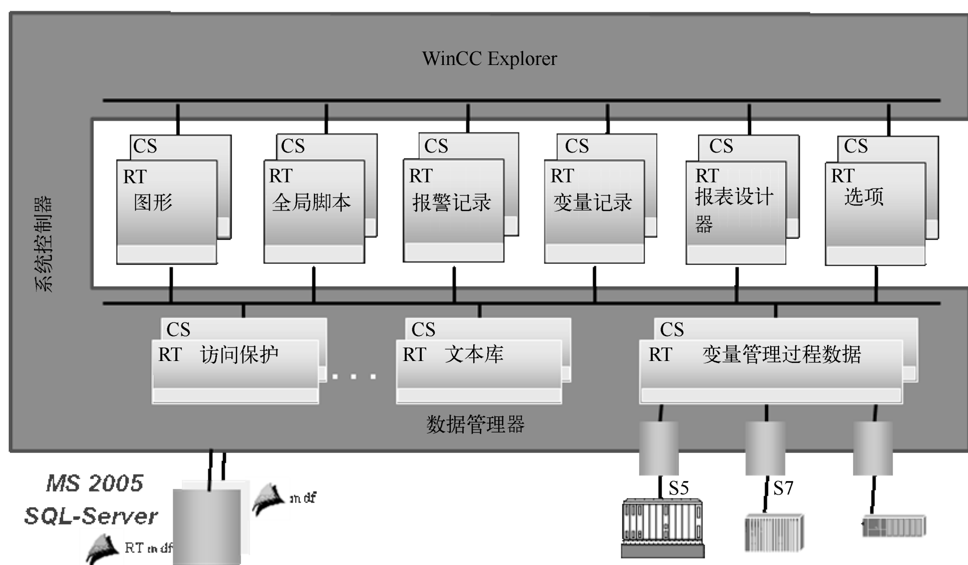



图 1-2　WinCC 体系结构示意图

图 1-2 中，系统控制器负责管理 WinCC 各站之间（如：HMI 即操作员站 OS，PLC 即自动化站 AS）的系统通信。而对于客户机/服务器的 WinCC 项目（即多用户系统或分布式系统），系统控制器还负责提供整个网络范围的项目视图，以及管理客户机和服务器之间的通信。数据管理器（或称变量管理器）是 WinCC 项目中处理中心任务的起点，与系统控制器密切协作，主要任务是进行变量的管理，通过通信通道访问过程数据。

## 1.4.1　WinCC 基本系统

WinCC 基本系统是 WinCC 的核心，它由以下子系统组成：

### 1. 变量管理器

变量管理器（Tag Management）实际上就是图 1-2 中的数据管理器，用于管理 WinCC 中所使用的外部变量、内部变量和通信驱动程序。

### 2. 图形编辑器

图形编辑器（Graphics Designer）用于设计 HMI 系统的各种图形画面，实现过程可视化。

### 3. 报警记录

报警记录（Alarm Logging）负责采集和归档报警消息。报警记录分为组态系统和运行系统两个组件。报警记录的组态系统称为报警记录编辑器，用于组态报警的种类、报警的内容、报警的时间等，其组态决定了运行系统的显示内容和显示形式；报警记录的运行系统主要负责过程值的监控、控制报警输出、管理报警确认等。

### 4. 变量记录

变量记录（Tag Logging）负责处理测量值，组态测量值的归档。

### 5. 报表编辑器

报表编辑器（Report Designer）用于设计报表的布局，提供了许多标准的报表布局，

也可由用户自己设计各种样式的报表布局，并可按照预定的时间打印报表。

6. 全局脚本

全局脚本（Global Script）允许设计人员设计脚本，以满足 WinCC 项目的需要。WinCC 的全局脚本系统提供 ANSI-C 和 VBScript 两种脚本。

7. 文本库

文本库（Text Library）用于编辑不同语言版本下的文本消息。

8. 用户管理器

用户管理器（User Administrator）用来分配、管理和监控用户对 WinCC 组态系统和运行系统的访问权限。

9. 交叉引用表

交叉引用表（Cross-Reference）负责搜索在画面、函数、归档和消息中所使用的变量、函数、OLE 对象和 ActiveX 控件等。

WinCC 基本系统由组态软件（Configuration Software，CS）和运行系统软件（Runtime，RT）组成。组态软件用于创建、设计和编辑 WinCC 项目，运行系统软件则用于在进行处理时执行项目，这样项目就“处于运行期”。

WinCC 基本系统按授权使用的过程变量（Power Tags，又称外部变量）的个数进行授权，按授权来分类，WinCC 基本系统分为完全版（RC）和运行版（RT）两个版本。完全版包括运行系统软件和组态软件的授权，运行版仅有运行系统软件的授权，WinCC 完全版和运行版都有 5 种授权规格：128 个、256 个、1024 个、8000 个（8K）和 65 536 个（64K）。

## 1.4.2　WinCC 选件

WinCC 选件允许用户扩展 WinCC 基本系统的功能，每个选件均需要一个专门的许可证（授权）。WinCC 选件主要包括：

1. WinCC/User Archive

用户归档选件，用于存储 WinCC 集成数据库中自由结构的数据记录。通过自由组态的 ActiveX 控件，数据记录可以在运行系统中以窗体或表格的形式显示。可以通过外部应用程序（如 Excel），对数据记录进行导入或导出，并做进一步处理。数据记录也可以作为配方或资料数据在 WinCC 和控制器之间交换。

2. WinCC/Server

服务器选件，用来将一个单用户系统解决方案扩展成最多可以有 12 个 WinCC 服务器，每个服务器最多可以有 32 个客户机的一种功能强大的（分布式）客户机/服务器系统。

3. WinCC/Redundancy

冗余选件，用于组态冗余系统，通过两个互连服务器的并行操作和故障事件中服务器间的自动切换，增强了 WinCC 和系统作为一个整体的可用性。

4. WinCC/Web Navigator

Web 浏览器选件，该选件允许用户借助于 Microsoft Internet Explorer 等浏览器，通过互联网（Internet）或企业内部网（Intranet）操作和监视整个过程，而不必对 WinCC 项目



做任何变动。瘦型客户机解决方案使用户除 PC 外，还可在 Windows CE 下使用可靠的本地设备和移动个人数字助手（PDA）来实现。

5. WinCC/ProAgent

过程诊断选件，可有选择地和快速地对工厂和设备进行过程诊断。由于它完全集成在 SIMATIC 过程诊断系统中，ProAgent 可提供基于 STEP 7、Engineering Tools 以及 SIMATIC S7 控制系统的一致性解决方案。

6. WinCC/Dat@Monitor

Dat@ Monitor 选件允许用户在任何办公用 PC 上，使用如 Microsoft Internet Explorer 或 Excel 那样的标准工具来显示和评估当前过程状态和历史数据，但不能操作过程。

7. WinCC/Connectivity Pack

连通性选件，该选件使其他的应用程序可以通过 OPC HDA 或 OLE-DB 访问 WinCC 归档系统中的数据，通过 OPC A&E 访问消息系统的数据。

8. WinCC/IndustrialDataBridge

IndustrialDataBridge 是一个可参数化的标准软件，支持通过 OLE-DB 和 OPC DA，将 WinCC 连接到外部数据库、办公应用程序和 IT 系统上。

9. SIMATIC IT PPA

使用 SIMATIC IT PPA 选件，可从不同数据源中读出过程和生产数据，将数据分组，以便随后的最优化分析和评估，并可将数据以压缩形式存储在 Microsoft SQL Server 数据库内的长期数据归档系统中。

10. WinCC/WinBDE

WinBDE 选件能保证有效的机器数据管理（故障分析和机器特征数据），其使用范围既可以是单台机器，也可以是整套生产设施。

11. WinCC/Audit

应用 Audit（操作和跟踪）选件，组态时采取相应措施（依照白皮书），SIMATIC WinCC 就可满足制药和食品工业（美国）的 FDA 要求。

12. WinCC/Logon

Logon 选件在管理员和用户两方面提供一系列安全机制，并以这种方式满足 FDA 的要求。

13. WinCC/IndustrialX

IndustrialX 选件允许用户利用 ActiveX 技术组态用户自定义的对象，该对象是标准化和可重用的，而且能统一修改。

14. WinCC/ODK

ODK 选件给出了开放性编程接口（C-API），使用户能用它来访问 WinCC 组态和运行时系统的数据和功能，甚至能建立用户自己的应用程序。

15. WinCC/CDK

CDK 选件用于通道驱动程序的开发。

### 1.4.3 WinCC 附加件

WinCC 附加件是由西门子公司的其他部门（如 WinCC Competence Center）和外部供

应商（如 WinCC Professionals and System Companies）开发和营销的。WinCC 附加件的支持由相关产品供应商提供。

WinCC 附加件能解决各种任务，诸如维护管理（MES 软件）、能源管理、导入过滤器、水管理的工业解决方案以及当在生产部门发生报警时，与其他制造商的 PLC 或无线呼叫的自动传送通信。

关于附加件以及相关制造商联系地址的更多信息，可在 Internet 上的附加件目录中找到。

# 第 2 章

# WinCC 的安装

本章以 WinCC V6.2 为例，介绍安装 WinCC 的硬件和软件要求、安装注意事项以及详细的安装步骤，针对安装过程中一些常见的问题和故障进行了分析，并提出了有效的解决方法。

## 2.1 WinCC 的安装要求

任何软件在开始安装之前，都需要首先了解该软件的安装要求，主要是了解软件安装的硬件要求和软件要求。WinCC 是以 PC 为硬件平台，基于 Microsoft Windows 2000/XP 的组态软件。因此，在安装 WinCC 之前，必须配置适当的硬件和软件，并保证它们正常运转。这意味着所有的硬件应该出现在 Windows 2000/XP 的硬件兼容性列表中，还意味着硬件和软件都必须被正确地安装和配置。WinCC 的安装要求包括安装的详细步骤和注意事项在其安装帮助文件“Installnotes. pdf”都有详细的介绍，该文件位于 WinCC 安装光盘的“PrintDocuments”文件夹下，建议读者尤其是初学者在安装 WinCC 前仔细阅读该文件。

### 2.1.1 安装 WinCC 的硬件要求

为了能可靠和高效地运行 WinCC，应满足一定的硬件条件，表 2-1 为安装 WinCC V6.2 的硬件要求。

表 2-1 安装 WinCC V6.2 的硬件要求

| 硬　件 | 最低要求 | 推荐配置 |
|---|---|---|
| CPU | 客户机：Intel Pentium Ⅲ，600 MHz<br>服务器：Intel Pentium Ⅲ，1 GHz<br>中央归档服务器：Intel Pentium 4，2 GHz | 客户机：Intel Pentium Ⅲ，1 GHz<br>服务器：Intel Pentium 4，2 GHz<br>中央归档服务器：Intel Pentium 4，2.5 GHz |
| 工作内存/RAM | 客户机：512 MB<br>服务器：1 GB<br>中央归档服务器：1 GB | 客户机：512 MB<br>服务器：>1 GB<br>中央归档服务器：≥2 GB |
| 硬盘上的可用存储空间<br>-用于安装 WinCC[①]<br>-用于使用 WinCC[②][③] | 客户机：500 MB/服务器：700 MB<br>客户机：1 GB/服务器：1.5 GB/中央归档服务器：40 GB | 客户机：700 MB/服务器：1 GB<br>客户机：1.5 GB/服务器：10 GB/中央归档服务器：在不同硬盘上的 2×40GB |

（续）

| 硬　　件 | 最 低 要 求 | 推 荐 配 置 |
|---|---|---|
| 虚拟内存④ | 1.5 倍速工作内存 | 1.5 倍速工作内存 |
| 用于 Windows 打印机假脱机程序的工作内存⑤ | 100 MB | >100 MB |
| 图形卡 | 16 MB | 32 MB |
| 颜色数量 | 256 | 真彩色 |
| 分辨率 | 800×600 | 1024×768 |

① 安装程序至少需要 100 MB 的可用内存空间，用于在安装操作系统的驱动器上的附加系统文件。通常操作系统位于驱动器“C:”。

② 取决于项目大小及归档和数据包的大小。一旦激活项目，至少必须有 100 MB 的自由内存空间可用。

③ WinCC 项目不应存储在压缩的驱动器或目录中。

④ 在区域“用于所有驱动器的交换文件总的大小”中为“指定驱动器的交换文件的大小”使用推荐的数值。请在“开始大小”域及“最大值”域中都输入推荐的数值。

⑤ WinCC 需要 Windows 打印机假脱机程序对打印机错误进行检测。因此，不能安装任何其他的打印机假脱机程序。

最低的硬件要求只能保证 WinCC 的运行，而不能保证在生产环境中满足大用户数、大数据量的访问。在实际配置时，用户应根据特定的应用需求，为 WinCC 配置适当的硬件。一般情况下，这些配置都会比表 2-1 中的最低要求高一些。对于单用户系统的运行，应满足表 2-1 中的最低硬件要求，如果需要系统高效地运行，则应满足推荐的配置要求，如果用户需要在线组态，则表 2-1 中的推荐配置为最低要求。

### 2.1.2　安装 WinCC 的软件要求

安装 WinCC 也应该满足一定的软件要求，在安装 WinCC 前就应该安装所需的软件并配置好。安装 WinCC 的软件要求主要包括操作系统及其补丁的要求、Internet 浏览器的要求、Microsoft 消息队列服务的要求以及 Microsoft SQL Server 的要求等。以下是安装 WinCC V6.2 时在这些方面的要求。

**1. 操作系统及其补丁**

（1）单用户系统和 WinCC 客户机

单用户系统和客户机/服务器系统的 WinCC 客户机推荐运行在 Windows 2000 Professional SP4 及以上版本或 Windows XP Professional SP2 及以上版本，也可以运行于 Windows Server 2003 标准版和企业版。

（2）WinCC 服务器

对于客户机/服务器系统的 WinCC 服务器，只能在 Windows Server 2003 标准版和企业版或 Windows Server 2003 R2 上运行。

另外，在安装 WinCC V6.2 的过程中，还将提示用户安装 Microsoft 的安全补丁。对于 Windows 2000 Professional SP4，其对应的 Microsoft 安全补丁为 KB828741；对于 Windows XP Professional SP2，其对应的 Microsoft 安全补丁为 KB319740。

**2. Internet 浏览器**

WinCC V6.2 要求安装 Microsoft Internet Explorer 6.0（IE6.0）SP1 或以上版本，IE6.0 SP1 随 WinCC V6.2 安装盘一起提供。安装和设置 IE6.0 必须选择以下选项：



- “安装选择”选项为“标准安装”。
- “更改 Windows 桌面”选项为“不更改”。
- “激活通道选择”选项为“无”。
- 如果需要使用 WinCC 的 HTML 帮助，则必须在 Internet Explorer 上进行设置。在 Internet Explorer 中单击菜单“工具”→“Internet 选项”→“安全”→“自定义级别”中开启 Java 脚本为“允许”。

3. Microsoft 消息队列服务

安装 WinCC 前，必须先安装 Microsoft 消息队列服务（Microsoft Message Queuing Services）。

4. Microsoft SQL Server

WinCC 的组态数据和运行时的归档数据使用关系数据库系统 Microsoft SQL Server 来存储，与 WinCC V6.2 对应的是 Microsoft SQL Server 2005。因此，在安装 WinCC V6.2 前必须先安装 Microsoft SQL Server 2005 SP1 Hotfix，并设置访问 SQL 服务器数据的相应访问权限。

## 2.2 WinCC 安装前的兼容性检查

在安装 WinCC 前，除了仔细阅读安装帮助文件“Installnotes.pdf”外，建议读者仔细阅读西门子公司发布的“WinCC 兼容性列表”，最大限度地避免 WinCC 安装完后出现运行不正常的问题，因为 WinCC 运行不正常多数是因为软件的兼容性问题导致的。另外，WinCC 是西门子 TIA 理念的重要组成部分，为保证 SIMATIC 产品在组态、编程、数据存储和通信等方面的一致性，检查 WinCC 与西门子其他产品（如 STEP 7）的兼容性是非常必要的。

“WinCC 兼容性列表”是随着 WinCC 版本的升级而不断更新的，列表中详细地列出了 WinCC 与操作系统、Internet 浏览器以及西门子其他软件产品之间的兼容性。读者可在西门子网站 www.ad.siemens.com.cn 的“支持中心”网页中输入“WinCC 兼容性列表”关键字搜索相应文档进行下载。随书光盘中附有西门子公司 2010 年 1 月发布的“WinCC 兼容性列表”。

## 2.3 WinCC V6.2 的安装

以下主要针对当前使用较多的 Windows XP Professional SP2 操作系统为例，详细介绍 WinCC V6.2 的安装。

### 2.3.1 安装注意事项

为了确保顺利地安装 WinCC V6.2 以及安装完成后 WinCC 能够正常运行，还需要注意以下事项：

1）确保以具有管理员权限的用户登录到 Windows。

2）确保 Windows 系统的计算机名和域-组策略已经设置完毕，WinCC 安装完成后不得再修改计算机名和域-组策略，否则 WinCC 将无法正常运行。

3）确保在安装 WinCC 前将 Windows 系统中运行的杀毒软件和防火墙暂时屏蔽，否则 WinCC 安装完成后无法正常运行或 Windows 系统无法正常运行。

4）确保将 WinCC 软件安装在不带任何中文字符的路径中，否则 WinCC 安装完成后 WinCC 无法正常运行。

5）确保安装 WinCC 前安装经西门子授权的 Microsoft SQL Server 2005，否则 WinCC 将无法安装。

## 2.3.2 安装步骤

在 Windows XP Professional SP2 操作系统以及 Internet Explorer SP1 安装完成，满足安装 WinCC V6.2 的软件要求的前提下，应该按以下顺序安装 WinCC V6.2 ：

1）安装 Windows Message Queuing Services。

2）安装 Microsoft SQL Server 2005 SP1 Hotfix（该软件随 WinCC 安装光盘一起发送）。

3）安装 Windows XP 的安全补丁。

4）安装 WinCC V6.2。

5）安装授权。

另外，如果考虑将 WinCC 项目集成到 STEP 7 项目中，实现在 WinCC 中直接调用 STEP 7 项目中的变量以及集成诊断等功能，除了必须保证 WinCC V6.2 与 STEP 7 兼容外，还必须确保在安装 WinCC V6.2 之前安装 STEP 7。

1. Microsoft Message Queuing Services 的安装

WinCC 使用了 Microsoft Message Queuing Services，虽然 Windows 2000 和 Windows XP Professional 操作系统都包含了消息队列服务组件，但并不是操作系统安装中默认安装的组件，因此必须单独安装。安装该组件需要相应的 Windows 安装盘。Windows XP Professional 下安装消息队列服务的步骤如下：

1）单击“开始”菜单→“控制面板”→“添加/删除程序”，弹出“添加/删除程序”对话框。

2）在“添加/删除程序”对话框中单击左边菜单条中的“添加/删除 Windows 组件”按钮，弹出“Windows 组件向导”对话框，如图 2-1 所示。

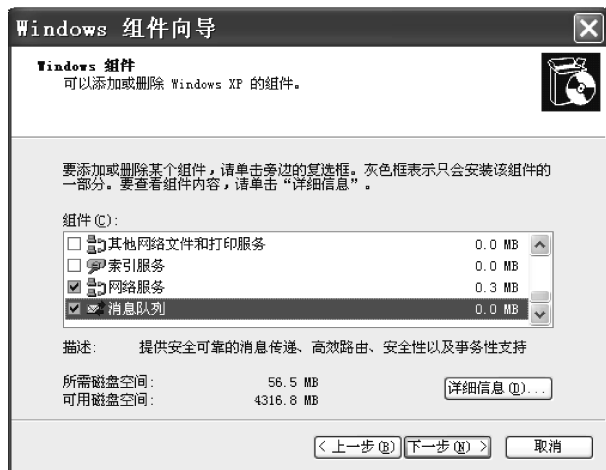



图 2-1 “Windows 组件向导”对话框



3）在“Windows 组件向导”对话框中激活“消息队列”，并单击“详细信息”按钮，打开“消息队列”对话框，如图 2-2 所示。

4）在“消息队列”对话框中，确保激活了“Active Directory 集成”、“触发器”和“公用”三个消息队列的子组件后，单击“确定”按钮，返回“Windows 组件向导”对话框。

5）在“Windows 组件向导”对话框中单击“下一步”按钮，开始安装消息队列服务。在安装过程中弹出“插入磁盘”对话框，如图 2-3 所示，插入所要求的 Windows 操作系统安装光盘，单击“确定”按钮，继续安装过程。

经过上述步骤，Microsoft 消息队列服务将安装到 Windows XP 系统中。

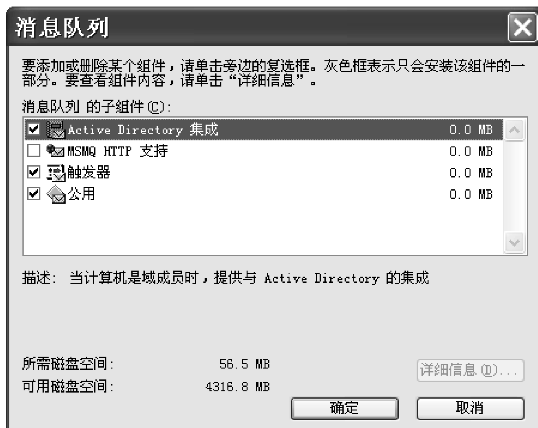



图 2-2 “消息队列”对话框

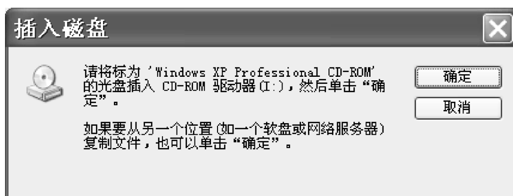



图 2-3 “插入磁盘”对话框

## 2. Microsoft SQL Server 2005 的安装

对于 WinCC V6.2，其对应的 SQL Server 版本为 Microsoft SQL Server 2005 SP1 Hotfix，该软件随 WinCC V6.2 软件一起提供。Microsoft SQL Server 2005 SP1 Hotfix 安装完成后，将建立一个新的名称为“WinCC”的 SQL Server 2005 实例，该实例总是使用英语进行安装。名称为“WinCC”的 SQL Server 2005 实例并不影响已经存在的 SQL Server 2005 实例，而且系统中即使已经安装了其他的 SQL Server 2005 实例，也必须安装名称为“WinCC”的 SQL Server 2005 实例，因为只有该实例才能和 WinCC 系统连接。Microsoft SQL Server 2005 SP1 Hotfix 的安装步骤如下：

1）将“Microsoft SQL Server 2005 SP1 Hotfix”的安装光盘放入光驱中，打开该光盘，双击可执行文件“Setup.exe”，开启安装过程。如果安装之前没有重新启动 Windows 系统，则将弹出图 2-4 所示对话框，提示重新启动系统后再安装。单击“OK”按钮退出该对话框，手动重新启动 Windows 系统以后重新开启安装过程即可。




图 2-4 重新启动 Windows 系统提示对话框

2）等待片刻，将弹出图 2-5 所示的对话框，单击“Next”按钮，弹出图 2-6 所示的软件许可协议对话框。

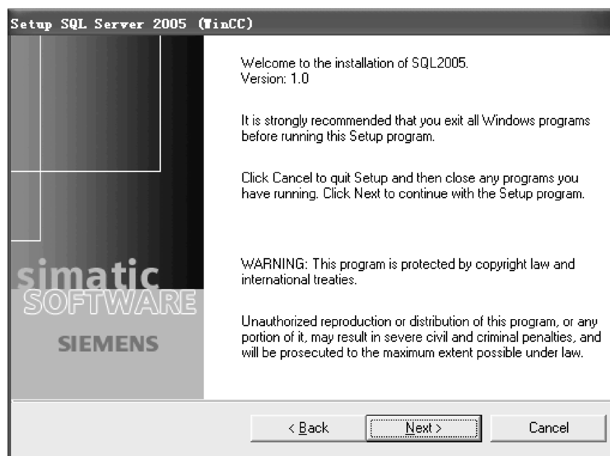



图 2-5　欢迎对话框

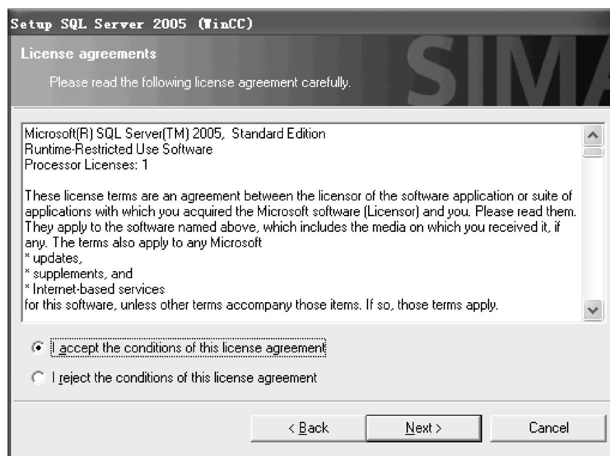



图 2-6　软件许可协议对话框

3）在图 2-6 所示对话框中激活“I accept the conditions of this license agreement”选项，单击“Next”按钮，弹出安装方式对话框，如图 2-7 所示。在该对话框中，用户可以选择“Package installation”（软件包安装）和“User-define installation”（用户自定义安装）两种安装方式，一般选择“Package installation”就可以了；用户还可以单击“Target directory”（目标文件夹）框内的“Browse”按钮，在弹出的对话框中选择软件的安装路径。设置完成后，单击“Next”按钮，打开下一个对话框。

4）在如图 2-8 所示的对话框中，激活“SQL Server 2005［WinCC］”软件包，单击“Next”按钮，弹出如图 2-9 所示的对话框。在该对话框中，显示了所有将要安装的软件包以及软件安装目标驱动器上的剩余空间，无需做任何设置。单击“Next”按钮，打开准备安装对话框，如图 2-10 所示。






图 2–7　安装方式对话框

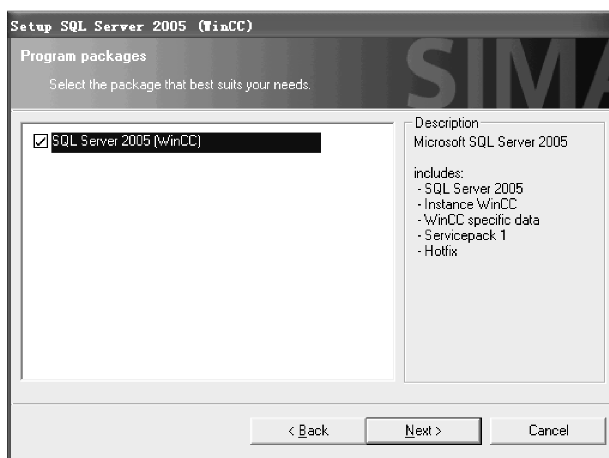



图 2–8　选择软件包对话框

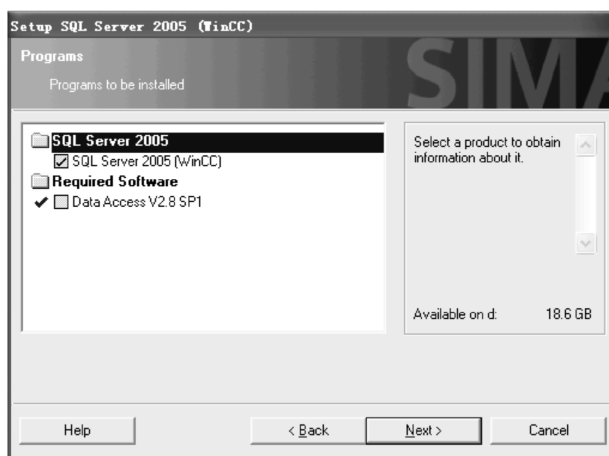



图 2–9　所有将安装的软件包

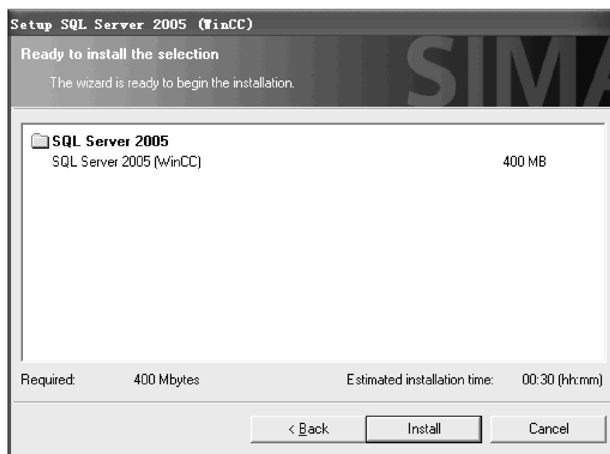



图 2-10　准备安装对话框

5）在图 2-10 所示对话框中，显示了安装“SQL Server 2005［WinCC］”软件包需要的存储空间以及花费的时间，同样无需做任何设置。单击“Install”按钮，将开始安装软件包。

6）开始安装软件包不久，安装程序将弹出图 2-11 所示的对话框。该对话框要求用户分配 SQL Server 的系统管理员口令，用户可以选择默认口令（Default password），也可以选择分配自己的指令。对于 SQL Server 的系统管理员口令，只允许使用 ASCII 字符且必须至少包含 14 个字符。本次安装选择默认口令，单击“Continue”按钮，将弹出开始复制文件对话框，如图 2-12 所示。




图 2-11　系统管理员口令设置对话框

7）在开始复制文件对话框中显示了当前的所有设置，单击“Install”按钮，软件安装正式开始，以后无需做任何设置。安装完成后，将弹出安装完成对话框，在该对话框中单击“Finish”按钮，关闭对话框，安装结束。



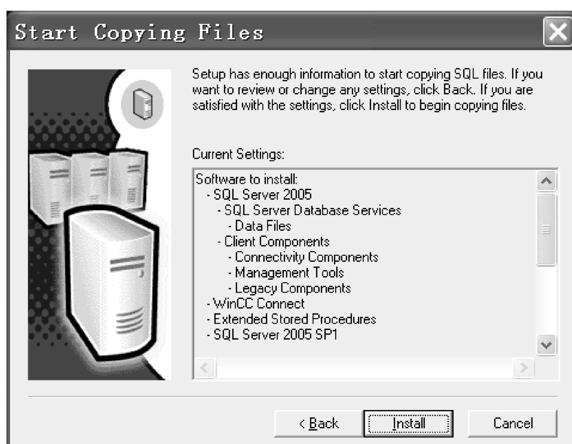



图 2-12　开始复制文件对话框

SQL Server 2005 SP1 Hotfix 安装完成后，还应该安装针对该软件的 MS 安全补丁，其软件名为“Microsoft SQL Server Native Client（9.00.2208.00）”，可以在 WinCC V6.2 的安装光盘中找到，详细路径为“\Tools\MS_SQLServer2005\sqlncli.msi”，双击“sqlncli.msi”，按提示进行安装就可以了。该补丁程序也可以在安装 WinCC 的过程中进行安装，在检查安装条件的阶段，如果先前没有安装 SQL Server 2005 的 MS 安全补丁，WinCC 安装程序将弹出图 2-13 所示的对话框，提示用户安装安全补丁程序，在该对话框中单击“是”按钮，按提示就可以完成该补丁程序的安装。

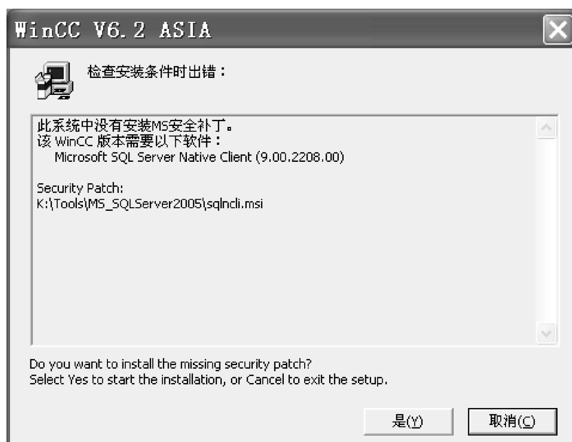



图 2-13　提示对话框

### 3. Windows XP 安全补丁的安装

由 2.1.2 节可知，对于 Windows XP Professional SP2 操作系统，其对应的 Microsoft 安全补丁为 KB319740。该补丁同样可以在 WinCC V6.2 的安装光盘中找到，详细路径为“软件\WinCC V6.2.2 ASIA\Tools\MS_XPSP2_KB319740\chs\WindowsXP-KB319740-v5-x86-CHS.exe”，双击“WindowsXP-KB319740-v5-x86-CHS.exe”，按照提示进行安装。

### 4. WinCC V6.2 的安装

在系统满足安装 WinCC 的软件要求后即可以进行 WinCC 的安装，其详细安装步骤

如下：

1）在光驱中插入“WinCC_V6.2_ASIA”光盘，双击“Start.exe”文件，弹出图 2-14 所示的安装界面，单击“安装 SIMATIC WinCC”，开启 WinCC 的安装。




图 2-14　WinCC V6.2 安装界面

在安装的开始阶段，WinCC 将根据其安装的软件要求逐一检查以下各项是否满足要求：

- 操作系统。
- 用户登录的权限。
- Internet Explorer。
- Microsoft 消息队列服务。
- Microsoft SQL Server。
- 是否重新启动操作系统（Windows Reboot）。

如果其中之一没有满足要求，则 WinCC 将停止安装，并在屏幕上弹出错误信息对话框，如图 2-15 所示，显示“OK”的项为满足要求，显示“!”的项不满足要求。

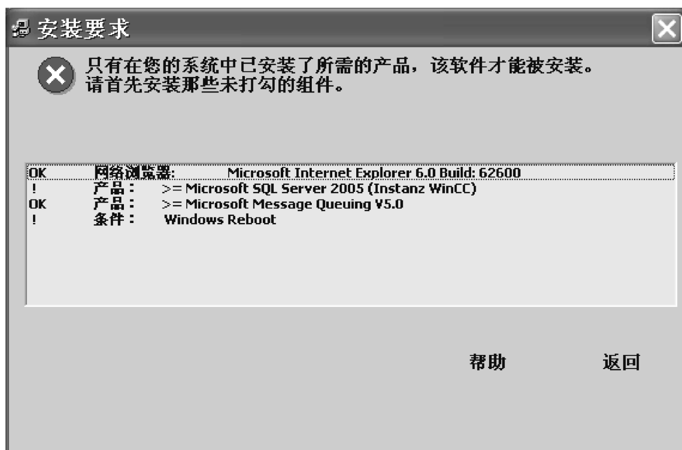



图 2-15　错误信息对话框



2）检查的各项均满足要求的情况下，将弹出如图 2-16 所示的对话框。

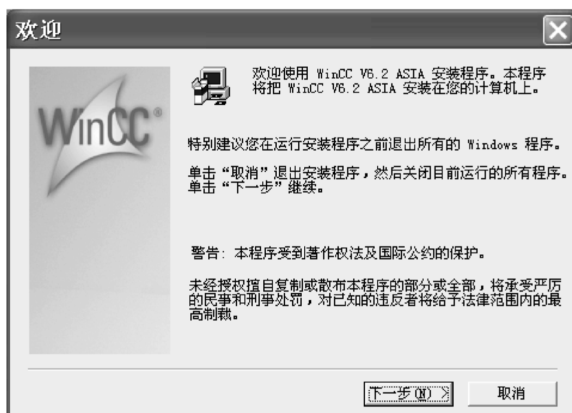



图 2-16　“欢迎”对话框

3）在该对话框中单击“下一步”按钮，在接下来显示的“软件许可证协议”对话框中单击“是”按钮，将弹出“用户信息”对话框，如图 2-17 所示。

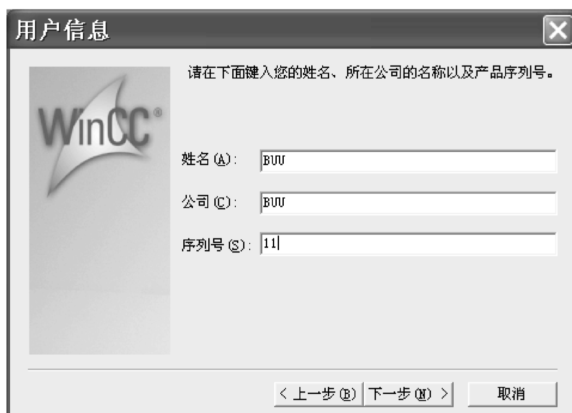



图 2-17　“用户信息”对话框

4）在“用户信息”对话框中，输入“姓名”、“公司”和“序列号”信息（序列号可任意输入），单击“下一步”按钮，弹出“选择目标路径”对话框，如图 2-18 所示。

5）“选择目标路径”对话框可以设置 WinCC 的安装路径，单击其对应的“搜索”按钮，在弹出的对话框中进行设置就可以了。设置好安装路径后，返回“选择目标路径”对话框，单击“下一步”按钮，将进入“选择附加的 WinCC 语言”对话框，如图 2-19 所示。

6）在“选择附加的 WinCC 语言”对话框中，除英语是默认安装的语言外，用户可以根据需要选择安装 WinCC 亚洲版支持的简体中文、繁体中文、日语和朝鲜语等多种语言（简体中文和繁体中文不能同时安装）。选择附加的安装语言后，单击“下一步”按钮，弹出“安装类型”对话框。




图 2-18　“选择目标路径”对话框

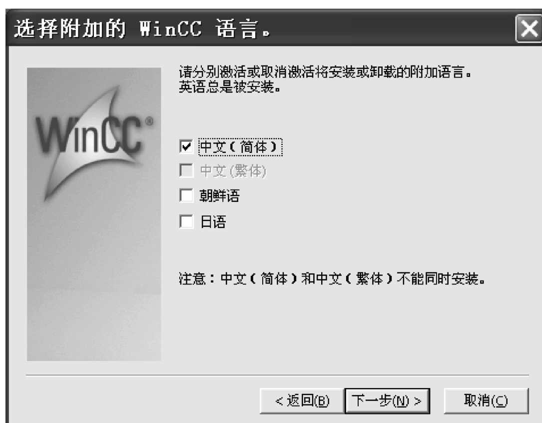



图 2-19　“选择附加的 WinCC 语言”对话框

7）“安装类型”对话框提供 3 种 WinCC 的安装选择：最小化安装、典型化安装和自定义安装。最小化安装包括 WinCC 运行系统、组态系统、SIMATIC 通信驱动程序和 OPC 服务器等组件的安装；典型化安装包括最小化安装的全部组件以及在自定义安装中默认激活的组件；自定义安装时用户可以自由选择需要的组件进行安装，“选择组件”对话框如图 2-20 所示，可以选择的组件及其描述见表 2-2。




图 2-20　“选择组件”对话框



表 2-2　WinCC 自定义安装可选组件表

| 名　称 | 描　述 | 可 选 组 件 |
|---|---|---|
| WinCC | 包含 WinCC 基本功能的组件 | RT（运行系统） |
| | | RC（组态系统） |
| | | 诊断 |
| 帮助 | 支持用户的 WinCC 应用程序文档 | 在线帮助 |
| | | WinCC 文档 |
| 通信 | 与各种 PLC 的通信驱动程序 | SIMATIC 设备驱动器 |
| | | 对象管理器 |
| | | AS-OS Engineering |
| | | STEP 7 符号服务器 |
| | | |
| OPC 服务器 | 世界标准和通用的用于过程连接的软件接口 | 数据访问 |
| | | 报警和事件 |
| | | 历史数据访问 |
| | | HDA 写访问 |
| | | XML Data Access |
| 选项 | 扩展基本功能的应用程序选项包 | 用户归档 |
| | | 冗余 |
| | | 服务器 |
| | | 基本过程控制 |
| | | 芯片卡 |

8）如果选择的是最小化安装或典型化安装，在“安装类型”对话框中单击“下一步”按钮；如果选择的是自定义安装，则在选择需要安装的组件后，在“选择组件”对话框单击“下一步”按钮，将弹出“授权”对话框，如图 2-21 所示。




图 2-21　“授权”对话框

9）在“授权”对话框中，显示了刚刚选择安装的组件需要的授权种类和授权方式，如果没有授权，WinCC 只能以演示模式运行，运行 1 小时后将自动退出。由于也可以在安装完成后进行授权，因此，选择“否，稍后进行授权”，单击“下一步”按钮，弹出“所选安装组态的概要”对话框，如图 2-22 所示。

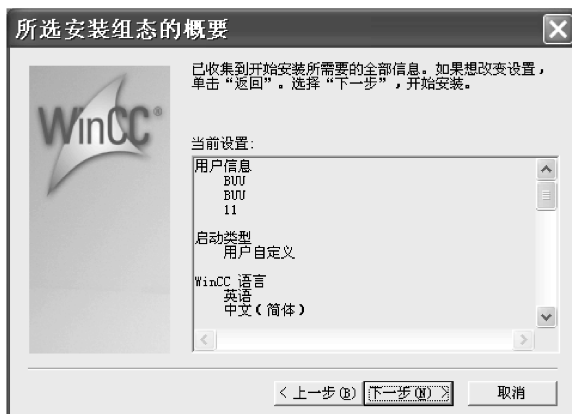



图 2-22　“所选安装组态的概要”对话框

10）“所选安装组态的概要”对话框列出了安装 WinCC 时的所有设置。如果要对这些设置进行修改，可单击“上一步”按钮；如果无需修改，单击“下一步”按钮。WinCC 安装程序将开始安装，将光盘上的文件复制到用户先前设置的目标路径下。

11）安装期间，将弹出“SIMATIC 安全控制”对话框，如图 2-23 所示，该对话框显示了针对 WinCC 定制的操作系统安全设置，用户必须单击“应用”按钮使这些设置生效。用户还可以单击“保存”按钮，将这些设置备份到文件中以备后续查看。设置的参数应用完成以后，单击“退出”按钮，继续安装过程。




图 2-23　“SIMATIC 安全控制”对话框

12）安装结束后，弹出“信息”对话框，如图 2-24 所示，提示用户备份 WinCC V6.0 版本的项目。单击“确定”按钮，在接下来的对话框中选择“是，我想现在重新启动计算机”，完成整个安装过程。



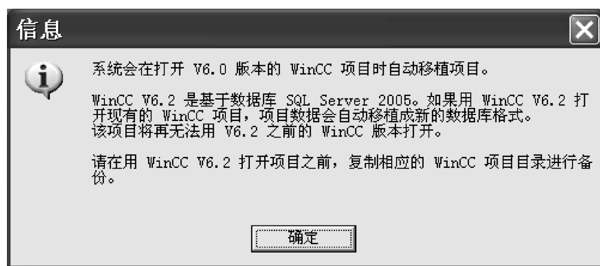



图 2-24 “信息”对话框

WinCC 安装光盘还提供了一些附加软件，用户可以根据需要选择安装。在图 2-14 所示的 WinCC 安装界面中单击“附加软件”，将打开附加软件的安装界面，如图 2-25 所示，单击需要安装的软件，按屏幕提示安装。WinCC 安装光盘提供了以下附加软件：

- 组态工具（Configuration Tool）。
- WinCC 归档组态工具（Archive Configuration Tool）。
- WinCC 文件服务器（WinCC File Serve）。
- 自动许可管理器（Automation License Manager）。
- 全局脚本实例项目（Global Script Example Project）。
- DXF Viewer Volo View Express 2.01（DXF 文件阅读器软件）。
- SQL Anywhere Studio 7（WinCC V5.0 和 V5.1 使用的数据库）。




图 2-25 附加软件安装界面

#### 5. 安装授权

为确保 WinCC 及其组件正常运行，必须安装授权，没有安装授权的 WinCC 只能以演示模式运行 1 小时，1 小时后 WinCC 将自动关闭，演示版只能使用 30 天，之后无法再激活。

安装授权是将软件授权从随 WinCC 软件提供的授权盘传递到计算机硬盘的过程，具体传递方法可参考软件产品证书。授权盘上设置有相应的安装计数器，授权从授权盘传递到计算机硬盘的过程中，安装计数器将减 1，如果安装计数器数值为 0，将不能再使用

此授权盘进行授权。授权也可以从计算机硬盘传递回授权盘，此时授权盘的安装计数器将加 1。

除了正常的授权以外，西门子还提供了一个紧急授权盘，紧急授权可以使用软件 14 天，用户可以在 14 天内再次获取正常的授权。

授权从授权盘传递到计算机硬盘后，授权文件保存在一个名为“AX NF ZZ”的隐藏文件夹中，此时切忌执行 SCANDISK、DEFRAG 或 VIRENSCAN 等操作，因为这些操作有可能损坏硬盘上的授权文件，导致授权丢失。

## 2.4　WinCC 的补充安装和卸载

### 1. WinCC 的补充安装

一旦系统中安装了 WinCC，当需要安装附加语言或添加组件时，就可以按以下步骤进行补充安装：

1）打开操作系统“开始”菜单，选择“设置”→“控制面板”→“添加/删除程序”。

2）在“添加/删除程序”窗口的左边选择“更改或删除程序”，在“当前安装的程序”列表中选择“SIMATIC WinCC ASIA V6.2”，并单击“更改/删除”按钮，启动 WinCC 安装程序。

3）WinCC 安装程序将打开图 2-26 所示的“安装/卸载”对话框。如果要安装附加语言，单击“选择用于安装或卸载的语言”，将弹出“选择附加的 WinCC 语言”对话框（如图 2-19 所示）；如果要添加组件，单击“选择安装或卸载的组件”，将弹出“选择组件”对话框（如图 2-20 所示）。在各自的对话框中选择要安装的单个组件或语言，单击“下一步”按钮。

4）将 WinCC 光盘放入光盘驱动器，按照屏幕上的提示进行后面的操作。

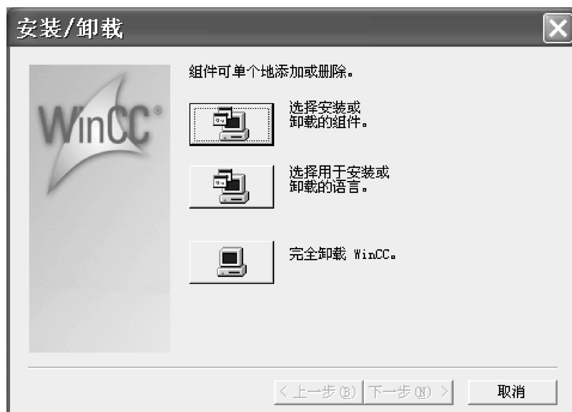



图 2-26　“安装/卸载”对话框

### 2. WinCC 的卸载

WinCC 的卸载可以是完整卸载，也可以是删除单个组件或语言，选择删除单个组件或语言时需要在光盘驱动器中插入 WinCC 安装光盘。WinCC 卸载的操作步骤和 WinCC 的补充安装的操作步骤类似，只需在图 2-26 所示的对话框中进行相应的操作即可，不再



详述。

3. SQL Server 2005 的卸载

完整卸载 WinCC 后，SQL Server 2005 WinCC 实例也必须卸载，打开“开始”菜单，选择“设置”→“控制面板”→“添加/删除程序”，在“添加/删除程序”窗口的左边选择“更改或删除程序”，在“当前安装的程序”列表中选择“Microsoft SQL Server 2005 (WinCC)”，并单击“删除”按钮，启动卸载程序。

## 2.5 WinCC V6.2 安装的常见问题及解决办法

### 2.5.1 Windows XP 无法安装消息队列

1. 问题现象

在 Windows XP 下安装消息队列时提示以下错误信息：

- 无法启动 MSMQ 服务。
- 错误代码：0x42c。
- 错误描述：依存服务或组无法启动。

2. 问题原因

消息队列服务依存的服务 MS DTC 服务（Microsoft Distributed Transaction Coordinator 分布式事务协调程序）没有启动且无法手动启动。

3. 解决办法

重新安装 MS DTC 服务，具体步骤如下：

1）打开 Windows 操作系统“开始”菜单，选择“运行”，在弹出的“运行”对话框中输入“cmd”，按〈Enter〉键或单击“确定”按钮，打开 DOS 命令窗口。

2）在 DOS 命令窗口依次输入以下 DOS 命令并按〈Enter〉键：

- net stop msdtc。
- msdtc-uninstall。
- msdtc-install。
- net start msdtc。

3）关闭 DOS 命令窗口。

### 2.5.2 安装时提示无法找到 ssf 文件

1. 问题现象

双击“Start.exe”文件，开启 WinCC 安装后，无法弹出图 2-14 所示的安装界面，提示错误信息“无法找到 ssf 文件”（No ssf File Found）。

2. 问题原因

实际上，“无法找到 ssf 文件”的问题不仅仅在安装 WinCC 的时候出现，在安装西门子的其他软件如 STEP 7、PCS 7、WinCC flexible 等软件的时候也出现。出现该问题的原因有两个可能：

① 软件安装的目标路径中存在中文字符。

② 将软件安装包复制到计算机硬盘进行安装时，软件安装包所在文件夹的层级过多。

3. 解决办法

① 针对“无法找到 ssf 文件”问题的第一种可能原因，由于 WinCC 以及西门子其他软件在对中文的支持上有一定的缺陷，因此，建议用户在安装西门子系列软件时安装路径不要包含中文字符。

② 针对“无法找到 ssf 文件”问题的第二种可能原因，建议用户将软件安装包复制到硬盘分区的根目录中或将软件安装包刻录成安装光盘再安装。

### 2.5.3　安装时反复要求重新启动计算机

1. 问题现象

包括 WinCC 在内，西门子的软件在安装前都要求重新启动计算机，但是经常出现的问题是重新启动计算机后又提示重新启动，再次重新启动后仍然出现相同的提示，无法进行软件的安装。对于 WinCC 来说，就是重新启动计算机后，启动 WinCC 的安装时，仍然出现图 2-15 所示的错误信息对话框，且“Windows Reboot”项不符合安装要求（显示为“!”）。

2. 问题原因

注册表中 PendingFileRenameOperation 键值存在安装程序暂挂项目，使得 Windows 系统认为存在未完成安装的安装程序（尤其是需要重新启动计算机完成安装的软件）。

3. 解决办法

删除注册表的 PendingFileRenameOperation 键值，具体操作步骤如下：

1）打开 Windows 操作系统“开始”菜单，选择“运行”，在弹出的“运行”对话框中输入“regedit”，如图 2-27 所示，并按〈Enter〉键或单击“确定”按钮，进入“注册表编辑器”。




图 2-27　“运行”对话框

2）在“注册表编辑器”的左边窗口中逐级打开“HKEY_LOCAL_MACHINE\ SYSTEM\ CurrentControlSet\ Control\ Session Manager”项，在右边窗口中找到“PendingFileRenameOperation”键值，如图 2-28 所示。

3）选中“PendingFileRenameOperation”键值，单击鼠标右键，在弹出的菜单中单击“删除”按钮并在弹出的对话框单击“确定”按钮，完成该键值的删除。

4）关闭“注册表编辑器”。

经以上步骤删除“PendingFileRenameOperation”键值后，无需再重新启动计算机，直接开始安装 WinCC 就可以了。



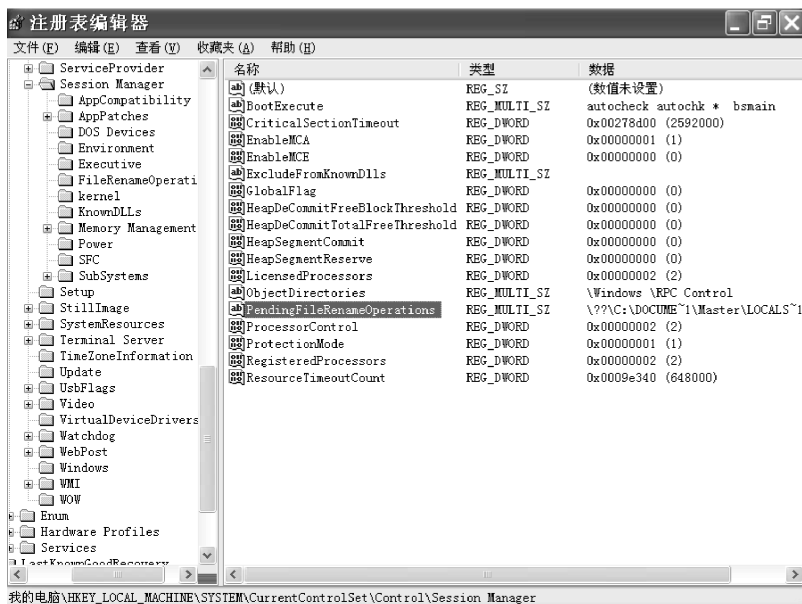



图 2-28 注册表 PendingFileRenameOperation 键值

需要用户注意的是，注册表的操作不当可能导致 Windows 系统无法使用，因此，用户必须慎重。

## 2.5.4 安装 WinCC 后 Windows 系统无法正常登录

### 1. 问题现象

安装 WinCC 后重新启动计算机，在 Windows 登录界面输入正确的计算机账户名和账户密码后，无法进入 Windows 系统桌面，系统自动注销并返回到 Windows 登录界面。

### 2. 问题原因

有两种可能的原因：

① 注册表的“Userinit”键值被删除。

② 系统文件夹“system32”下的“userini. exe”文件被删除。

导致上述两种可能的原因是由于在安装 WinCC 前没有退出或屏蔽系统中已经安装的与 WinCC 不兼容的杀毒软件和防火墙（如瑞星的杀毒软件和防火墙、卡巴斯基反病毒软件等）。因为 WinCC 安装程序会改变 Windows 系统的登录方式，屏蔽“使用欢迎屏幕”的登录方式（该方式单击账户名进行登录），改为传统的“使用登录对话框”的登录方式（该方式需输入账户名），而杀毒软件和防火墙在阻止 WinCC 安装程序改变 Windows 系统登录方式这种行为的过程中，可能误伤注册表的“Userinit”键值或系统文件“userini. exe”。

从以上分析可以看出，避免“安装 WinCC 后 Windows 系统无法正常登录”这个问题出现的关键是在安装 WinCC 前将 Windows 系统中已经安装的杀毒软件和防火墙退出或屏蔽，或者对于一个全新安装的系统，选择先安装 WinCC 等西门子软件，然后再安装杀毒软件和防火墙。另外，为保险起见，用户可以选择在安装 WinCC 前备份注册表或创建一

个系统还原点，这样当问题真的出现时可以尽快地恢复系统。

3. 解决办法

当“安装 WinCC 后 Windows 系统无法正常登录”这个问题真的出现后，有以下三种有效的解决办法：

① 重新安装 Windows XP 系统和 WinCC。该方法虽然有效，但是意味着一切从零开始、工作量大、耗时长、效率低，而且必须冒着再次出现同样问题的风险，因此该方法并不可取。

② 使用 Windows XP 故障恢复控制台恢复 Windows 并重新安装 WinCC。该方法同样有效，且无需重新安装 Windows，但是如果安装 WinCC 前没有备份注册表，则注册表只能恢复到 Windows 刚安装完后的注册表，很多应用程序和驱动程序无法正常使用，工作量同样比较大，而且该方法涉及很多 DOS 指令，比较复杂，要求用户掌握很深的计算机知识，重新安装 WinCC 也冒着再次出现同样问题的风险，因此该方法也不可取。

③ 使用 WinPE 修复 Windows 的注册表或系统文件。该方法非常有效，修复注册表或系统文件后，Windows 系统可以正常启动，其应用程序包括 WinCC 均无需重新安装即可正常使用、工作量小、效率高、且比较简单，适合大多数用户采用。

从以上分析可知，采用第三种方法能够有效地解决“安装 WinCC 后 Windows 系统无法正常登录”的问题，且方法简单、工作量小、效率高。WinPE 是近几年出现的 Windows 系统修复工具，具有很强的功能，最常用的是“深山红叶”WinPE。该方法的具体操作步骤如下：

1）计算机重新启动，进入 CMOS 设置界面将“First Boot”选项设置为光盘启动。

2）用“深山红叶”WinPE 光盘启动进入 WinPE 光盘操作系统。

3）在光盘操作系统中单击“开始”菜单 →“强力系统修复 ERD2003”→“设置当前系统目录”（必须设置该步骤，例如操作系统在驱动器 C：，则系统目录为 C：\windows）→ 单击“确定”按钮。

4）单击“开始”菜单 →“强力系统修复 ERD2003”→“注册表编辑器”（注意这里打开的一定是计算机本机的注册表编辑器，而不是光盘操作系统的注册表编辑器）

5）在“注册表编辑器”左边窗口中逐级打开“HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon”项，并在右边窗口中查看有无“Userinit”键值。如果没有，则在右边窗口的空白处单击鼠标右键→“新建”→“字符串值”→字符串值命名为“Userinit”→单击鼠标右键将“Userinit”的数据设置为“C：\WINDOWS\system32\userinit. exe”。

6）如果“C：\WINDOWS\System32”下的“userinit. exe”文件被删除了，则可以在 WinPE 光盘操作系统中将一个正常 Windows 系统的“userinit. exe”文件复制到本计算机的“C：\WINDOWS\System32”路径下。

7）退出 WinPE 光盘操作系统，重新启动计算机并选择从硬盘启动即可正常进入本机的 Windows 系统。

### 2.5.5　Windows XP SP3 安装 WinCC V6.2 时无法安装 KB319740

1. 问题现象

在 Windows XP SP3 系统中安装 WinCC V6.2 时提示没有安装安全补丁 KB319740，但



在安装 WinCC 光盘提供的 KB319740 时弹出如图 2-29 所示的错误信息，KB319740 无法安装，也导致无法安装 WinCC V6.2。




图 2-29　KB319740 安装错误信息

2. 问题原因

实际上，查看“WinCC 兼容性列表”可知，除了 WinCC V6.2 SP3 与 Windows XP SP3 兼容外，WinCC V6.2 SP2、WinCC V6.2 以及 WinCC V6.0 的所有版本均与 Windows XP SP3 不兼容。由于 Windows XP SP3 系统中不需要安装 KB319740 安全补丁（由其他补丁代替），而 WinCC V6.2 SP2 和 WinCC V6.2 在安装的开始阶段必须检查系统中是否安装 KB319740，因此用户在 Windows XP SP3 系统中安装 WinCC V6.2 SP2 和 WinCC V6.2 时会遇到上述问题。

3. 解决办法

虽然因为 WinCC V6.2 SP2、WinCC V6.2 与 Windows XP SP3 不兼容在安装时遇到了上述问题，但并不代表在 Windows XP SP3 系统中无法成功安装 WinCC V6.2 SP2 和 WinCC V6.2。用户可以通过修改系统注册表，在注册表的相关位置创建 KB319740 的信息，满足 WinCC V6.2 安装程序的检查要求的方法来解决，具体操作步骤如下：

1）打开 Windows 操作系统“开始”菜单，选择“运行”，在弹出的“运行”对话框中输入“regedit”，并按〈Enter〉键或单击“确定”按钮，进入“注册表编辑器”。

2）在“注册表编辑器”的左边窗口中逐级打开“HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates\Windows XP\SP3”项。

3）用鼠标右键单击“SP3”→“新建”→“项”→设置项名称为“KB319740”（无需输入引号）。

4）关闭“注册表编辑器”。

通过上述操作在注册表中创建了 KB319740 项后，就可以通过 WinCC V6.2 安装程序针对 KB319740 安全补丁进行检查了，在其他安装要求均满足的情况下也就可以实现 Windows XP SP3 下安装 WinCC V6.2 了。

实际上，还是 Windows XP SP2 与 WinCC V6.2 的兼容更好，因此，如果用户没有什么特殊需求，则建议优先选择 Windows XP SP2 作为系统平台。

# 第 3 章

# WinCC 的项目管理和变量管理

本章主要介绍 WinCC 的系统架构、项目类型、在项目管理器中新建或打开 WinCC 项目，以及在项目中通过变量管理器组态 WinCC 通信和 WinCC 变量等内容。针对 WinCC 项目创建或运行中出现的一些问题，本章进行了详细分析并提出了有效的解决方案。

## 3.1　WinCC 的项目管理

WinCC 以项目的形式来实现 HMI/SCADA 系统的组态，系统所需要的所有数据、参数和设置等都必须在由 WinCC 项目管理器创建的项目中进行统一管理。使用 WinCC 组态一个项目的基本任务和步骤如下：

1）启动 WinCC 并新建一个 WinCC 项目。

2）建立 WinCC 和控制器（通常是 PLC）的通信连接。

3）定义监控变量。

4）创建和编辑过程画面。

5）组态消息系统。

6）组态变量记录。

7）组态报表（如果系统需要）。

8）组态 WinCC 运行系统的属性。

9）激活 WinCC 项目。

10）测试项目。

其中，第 1）~3）步必须严格按照顺序执行，后续的组态必须在前三步工作的基础上进行。第 4）~7）步工作可根据实际情况顺序、交叉或并行执行，而激活项目和测试项目可以根据需要在组态的过程中多次执行。

### 3.1.1　WinCC 的系统架构

WinCC 是一个模块化、可延展的系统，扩展能力很强，因此 WinCC 的系统组态非常灵活，既适用于简单的单用户系统，又适用于复杂的多用户解决方案。WinCC 常用的系统架构有单用户系统（Single-User System）、多用户系统（Multi-User System）和分布式系统（Distributed System），还可以根据需要实现冗余系统（Redundancy System）、中央归档服务器（Central Archive Server）和基于 Web 访问的 IE 客户端（Web-based Client）等系



统的组态。

1. 单用户系统

单用户系统是最简单的系统架构，适合配置于小型的应用，但具备独立的控制和可视化系统组件。组态单用户系统应该在 WinCC 中创建单用户项目，运行该项目的站在系统中称为操作站（Operation Station，OS）。操作站执行系统中的所有任务，具有本身的过程通信，本身的画面和归档，还可以有各种选件用于连接到自动化系统（Automation Station，AS）。另外，单用户系统中的单用户 WinCC 项目还可以作为一个 OPC 服务器与其他系统通信、传送数据。典型的单用户系统如图 3-1 所示。

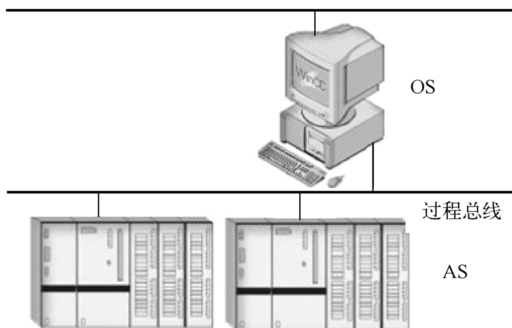


图 3-1　单用户系统

图 3-1 中，过程总线用于将单用户系统连接到自动化系统，组态时可以选择基于 MPI（Multi-Point Interface）、PROFIBUS-DP 和 Industrial Ethernet（工业以太网）等协议的总线。

2. 多用户系统

多用户系统由一台服务器和多个客户机（操作站）组成。通常对于小型系统，即数据不需要分布到多个服务器的情况下，组态带有过程驱动器连接的单服务器。该服务器负责实现所有公共的功能：连接自动化系统、协调客户机以及为客户机提供过程值、归档数据、消息、过程画面和协议等。多个客户机通过过程驱动器连接访问服务器上的项目，这些客户机协同运行，各自可以执行相同的或不同的任务。因此，对于多用户系统，只需组态服务器项目（对应 WinCC 的多用户项目），无需组态客户机项目。

多用户系统可以实现在不同的客户机上显示与同一过程相关的不同信息。例如，用户可以使用一个客户机来显示过程画面，第二个客户机则专门用于实现显示和确认消息的功能。这样的客户机既可以并排布置，也可以位于完全不同的位置，其数据完全由服务器提供。多用户系统还可以实现操作控制来自多个不同位置的过程，例如，沿生产线的不同位置的过程。

典型的多用户系统如图 3-2 所示。多用户系统的运行基于客户机/服务器的原理，服务器承担集中任务，每增加一台客户机都将增加服务器的工作负载，因此系统最多可以有 32 个客户机。服务器必须安装 Microsoft Windows 2000/2003 服务器版操作系统，WinCC 除需安装基本系统的授权外，还需 WinCC Option 服务器版的授权。

3. 分布式系统

WinCC 可以组态复杂的分布式系统，用于处理大量数据的大型系统。在分布式系统中，整个项目的任务或应用程序可以分布在几台服务器上，这样可以扩大项目的功能范围，减轻每台服务器的负担，并保证良好的工作性能。分布式系统的结构示意图如图 3-3 所示。原则上，在以下情况下，将整个可视化任务分配在几个服务器上是合理的，例如，可以按照系统的物理结构（如地铁系统以站为单位，工厂以车间为单位）分配服务器，也可以按功能（如消息服务器、归档服务器等）进行分配。

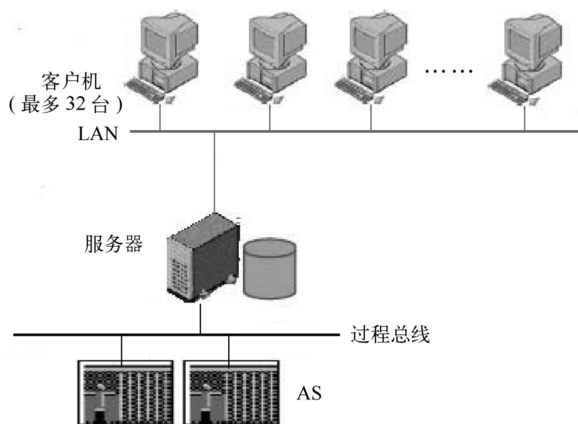



图 3-2　多用户系统

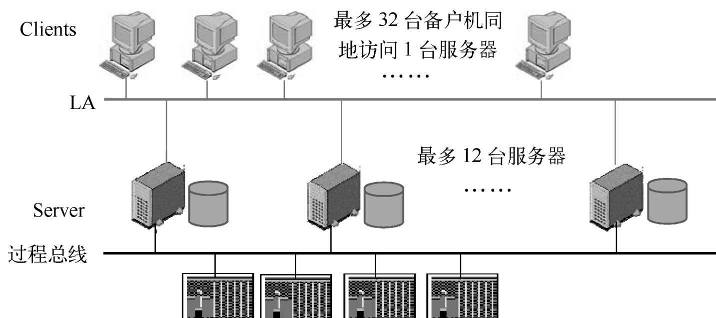



图 3-3　分布式系统

分布式系统的主要特点是一个或多个客户机访问多个服务器，服务器也可以以客户机的角色访问另一台服务器的数据。在运行期间，分布式系统中的每台客户机均可显示多达 12 个不同服务器或冗余服务器对中的数据。分布式系统中的各个客户机将使用基准画面和某些本地数据单独进行组态。用于显示过程数据的服务器数据从服务器传送到客户机，并可在必要时自动进行更新。

分布式系统中的每个服务器需要一个服务器授权（WinCC/Server）。系统的实际分布是通过相应的组态完成的，完整视图由客户机来实现，这些客户机同时访问不同服务器项目的数据。根据需要，还可以为每台服务器构造冗余服务器（需要 WinCC/Redundancy 授权）。

### 3.1.2　WinCC 项目的类型

在 WinCC 中可以创建的项目类型分为：单用户项目、多用户项目和客户机项目。单用户项目用于实现单用户系统，多用户项目用于创建多用户系统或分布式系统的服务器项目，而客户机项目用于创建分布式系统的客户机项目。

对于不同的系统架构需要设置不同的工程项目类型、WinCC 选件、组建以及授权，用户在创建项目时，可以根据实际需求选择项目类型，也可以在创建项目后在“项目属性”中更改项目的类型。



1. 单用户项目

前面已经提到，单用户项目用于实现单用户系统，整个系统中只有一台计算机进行工作，运行 WinCC 项目的该计算机在系统中称为操作站（OS），实际上是进行数据处理的服务器和操作员输入站。其他计算机不能通过 WinCC 访问该项目。

实际上，单用户项目在自动化网络系统中，除了在监控级只有一台计算机作为 WinCC 项目服务器而不存在项目客户机，其他的，例如与控制级的通信连接等，与多用户项目没有区别。

2. 多用户项目

如前面所述，多用户项目用于创建多用户系统或分布式系统的服务器项目。如果系统架构为多用户系统，则无需在客户机上创建单独的客户机项目。如果系统是具有多个服务器的分布式系统，则必须在客户机上创建单独的客户机项目，这种情况同样适用于只想访问一个服务器但又需要客户机上的附加组态数据的情况。

3. 客户机项目

如果在 WinCC 中创建了多用户项目，则随后必须创建对服务器进行访问的客户机，并在用做客户机的计算机上创建一个客户机程序。

如果组态的是多用户系统的客户机，则该客户机只访问一台服务器。由于 WinCC 项目在服务器上，所有的数据也在服务器上，客户机上没有单独的客户机项目，因此，必须在服务器的 WinCC 项目中将客户机添加到该项目的计算机列表中，客户机的计算机属性也是在服务器上进行组态的。

如果组态的是分布式系统的客户机，即该客户机可以访问系统中的多台服务器，则应该在客户机上创建一个客户机项目，并组态其项目属性。由于分布式系统中系统的组态数据和运行数据分布在不同的服务器上，因此需要在这些服务器上创建各自的数据包，并将其数据包自动或手动装载到需要访问其数据包的客户机上。该装载过程只需一次，如果服务器上的组态数据被修改了，则 WinCC 可以自动更新对应的数据包，并将更新的数据包自动下载到已经装载过此数据包的客户机上。另外，分布式系统的客户机项目还可以保存客户机本身的组态数据，如过程画面、脚本和变量等。

### 3.1.3 WinCC 项目的创建

在创建 WinCC 项目前，应该对项目进行初步的规划，以确定项目的组态方式、项目类型、项目路径等。在开始规划项目的时候，应该已经确定整个系统的架构，即采用的是单用户系统、多用户系统还是分布式系统，然后明确当前创建的项目类型是单用户项目、多用户项目还是客户机项目，最后要确保持有所需的相应授权。

在明确创建 WinCC 项目的以上信息后，就可以开始创建新的项目了。

1. 启动 WinCC

用鼠标双击 Windows 桌面的“SIMATIC WinCC Explorer”图标或单击“开始”→“所有程序”→“SIMATIC”→“WinCC”→“WinCC V6.2 ASIA”可以启动 WinCC，进入 WinCC Explorer 即 WinCC 项目管理器。

如果是首次启动 WinCC，将弹出“WinCC 项目管理器”对话框，如图 3-4 所示，用户可以通过该对话框开始创建新项目或打开已经存在的项目。如果不是首次启动 WinCC，

则 WinCC 会自动打开上次启动时最后打开的项目。如果希望启动 WinCC 项目管理器而不打开某个项目，则可以在启动 WinCC 的同时按〈Shift + Alt〉组合键并保持，直到出现 WinCC 项目管理器窗口。如果退出 WinCC 项目管理器前打开的项目处于激活状态（运行），则重新启动 WinCC 时将自动激活该项目（可通过同时按〈Shift + Ctrl〉组合键并保持取消自动激活）。




图 3-4　“WinCC 项目管理器”对话框

### 2. 新建项目

首次启动 WinCC 可以自动开始新建项目的过程，也可以在 WinCC 项目管理器窗口的工具栏单击新建图标或菜单栏中选择“文件”→“新建”等方式开始新建 WinCC 项目。新建项目中最重要的工作是选择项目类型、设置项目名称和路径等。

项目类型在图 3-4 所示的“WinCC 项目管理器”对话框中进行选择，选择相应的项目类型后在对话框中单击“确定”按钮，将弹出图 3-5 所示的“创建新项目”对话框，在该对话框中设置项目的名称和路径。完成项目名称和路径的设置后单击“创建”按钮，一个新的 WinCC 项目就生成了，该项目基本上是一个空项目，用户可以在此基础上通过项目管理器进行进一步的组态。




图 3-5　“创建新项目”对话框

需要用户注意的是，为保证项目能在 WinCC 中正常运行，在设置项目的名称和路径时，应遵循以下原则：



1）项目的名称和项目的路径（包括各级文件夹的名称）不能包含中文字符。

2）尽量不要将项目创建在 WinCC 软件所在的文件夹和分区，建议用户创建一个单独的分区用于创建 WinCC 项目。

3）当 WinCC 项目需归档大量的数据时，应确保项目所在的分区具有足够的存储空间。

4）设置项目名称时，不能出现不允许的字符，例如空格、分号、问号等。

### 3. 设置项目属性

在新生成的 WinCC 项目的基础上或者在项目的组态过程中都可以对该项目的属性进行设置。在图 3-6 所示项目管理器的浏览窗口利用鼠标右键单击项目名称（图中为“Test”），在弹出的菜单中选择“属性”，即可进入“项目属性”对话框对项目的属性进行设置，如图 3-7 所示。




图 3-6 WinCC 项目管理器




图 3-7 “项目属性”对话框

“项目属性”对话框包含“常规”、“更新周期”、“热键”和“选项”四个选项卡：“常规”选项卡用于显示和编辑当前项目的一些常规信息，如项目类型、创建者、创建日期、修改者、修改日期等；“更新周期”选项卡中可以查看项目的画面窗口和画面对象可设置的更新周期，用户还可以自定义 5 个范围在 100ms～10h 的更新周期，如图 3-7 所示；“热键”选项卡中可以定义 WinCC 用户登录和退出以及硬拷贝等操作的热键（快捷键）；“选项”选项卡提供了一些附加的项目选项供用户选择，例如 ES 上允许激活、使用激活的 XP 用户界面设计等。

**4. 设置计算机属性**

和项目属性一样，用户可以选择在刚生成新项目时或在项目进一步的组态过程中设置计算机属性。

如果当前项目的项目类型为多用户项目，如果在创建项目时没有添加访问服务器项目的客户机或还需要添加新的客户机，可以在图 3-6 所示项目管理器的浏览窗口用鼠标右键单击“计算机”，并在弹出的菜单中选择“添加新计算机”进行添加，但必须保证添加的计算机与客户机的计算机物理名称一致。在多用户项目中，必须单独为每台创建的计算机（包括服务器和客户机）设置属性。

设置计算机属性的方法是：用右键单击项目管理器浏览窗口的“计算机”，在弹出的菜单中选择“属性”，弹出“计算机列表属性”对话框，如图 3-8 所示。在该对话框中，“计算机列表”显示了项目所创建的所有计算机，选择需要设置属性的计算机名称，并单击对话框中的“属性”按钮，将进入“计算机属性”对话框，如图 3-9 所示。




图 3-8　“计算机列表属性”对话框




图 3-9　“计算机属性”对话框

在“计算机属性”对话框中，包含了“常规”、“启动”、“参数”、“图形运行系统”和“运行系统”五个选项卡：

●“常规”选项卡用于显示或修改当前计算机的名称和计算机类型，并在列表中显示



了所有的客户机名称（如果当前计算机的类型是服务器）或服务器名称（如果当前计算机的类型是客户机）。

- “启动”选项卡中包含两个列表：“WinCC 运行时的启动顺序”列表用于设置项目运行时启动的 WinCC 功能模块，例如图形运行系统、变量记录运行系统、报警记录运行系统、报表系统等；“附加的任务/应用程序”列表可以设置项目运行时启动的附加任务/应用程序，例如用户可以选择在项目运行的同时启动 Excel。
- “参数”选项卡用于设置项目运行时的一些参数，这些参数将影响当前计算机的工作方式，主要包括运行时的语言设置、运行时的默认语言、禁止键（禁用 Windows 的典型组合键）、PLC 时钟设置、运行时显示时间的时间基准和中央时间和日期格式化等参数的设置。
- “图形运行系统”选项卡主要用于设置项目在当前计算机上的启动画面和窗口属性等，根据项目的实际情况，可对项目中的服务器和各个客户机设置不同的启动画面和窗口属性。该设置也可以在完成过程画面的创建后进行。
- “运行系统”选项卡用于设置项目运行时的 Visual Basic 脚本（包括图形脚本和全局脚本）的调试选项、监视器键盘（软键盘）、图片高速缓存以及鼠标指针的显示图标等信息。在此选项卡中的设置将影响当前计算机上的所有项目。

所有选项卡的相关参数设置完成后，单击对话框中的“确定”按钮，结束计算机属性的设置。

### 3.1.4 WinCC 项目的文件结构

新建一个 WinCC 项目后，会在图 3-5 所示对话框中设置的项目路径中自动生成一个与项目名称相同的文件夹，即项目文件夹，并在项目文件夹中自动生成一些与项目有关的文件和文件夹，用户可以通过这些文件和文件夹查找一些附加的信息。WinCC 项目文件夹的结构是动态的，取决于安装 WinCC 时安装的软件工具以及项目中激活的 WinCC 功能模块，表 3-1 列出了 WinCC 项目文件夹中一些常见的且比较重要的文件和文件夹。

表 3-1 WinCC 项目的文件结构

| 路 径 | 文 件 名 | 类 型 | 功 能 |
|---|---|---|---|
| 项目文件夹 | —— | 文件夹 | 包含为项目创建的全部文件 |
| \项目文件夹\GraCS | —— | 文件夹 | 包含项目所有过程画面和画面文件 |
| \项目文件夹\ | 项目名称.MCP | 项目文件 | 项目的主文件 |
| \项目文件夹\ | 项目名称.MDF | 项目文件 | 组态数据的数据库文件 |
| \项目文件夹\ | 项目名称 RT.MDF | 项目文件 | 运行数据的数据库和用户归档数据库文件 |
| \项目文件夹\ArchiveManager\AlarmLogging | 计算机名称_项目名称_ALG_起始时间_结束时间.MDF | 项目文件 | 运行系统数据报警记录文件 |
| \项目文件夹\ArchiveManager\TagLoggingFast | 计算机名称_项目名称_TLG_F_起始时间_结束时间.MDF | 项目文件 | 运行系统数据快速变量记录文件 |

（续）

| 路　径 | 文　件　名 | 类　型 | 功　能 |
|---|---|---|---|
| \项目文件夹\ArchiveManager \TagLoggingSlow | 计算机名称_项目名称_TLG_S_起始时间_结束时间.MDF | 项目文件 | 运行系统数据慢速变量记录文件 |
| \Siemens\WinCC\bin | WinCC_SQL.MDF | 数据库文件 | 空白的运行系统数据库文件文件,必要时可以将其复制到项目并重命名后应用,创建新项目是该文件自动创建 |
| \Siemens\WinCC\Diagnose | *.LOG | 日志文件 | 记录设定数据、系统状态和应用时的错误信息 |

## 3.1.5　WinCC 项目的激活

新建一个 WinCC 项目并按照上述方法完成项目属性、计算机属性等的设置后，就可以根据需要对项目进行进一步的组态，例如组态过程画面、变量记录、报警记录、报表系统等。在组态的过程中或所有组态完成后，为了测试组态的结果，需要激活（运行）项目。

### 1. 激活项目

在项目管理器中激活当前的项目有两种方法：

- 在项目管理器的菜单栏中用鼠标单击“文件”→“激活”。
- 在项目管理器的工具栏中用鼠标单击▶图标激活项目

另外，前文已经提及，当用户打开一个 WinCC 项目并在该项目处于激活的状态下关闭了 WinCC 项目管理器，则下一次启动 WinCC 时将自动打开并激活该项目。由于这种操作方式很容易导致 WinCC 处于死机状态，因此建议用户在关闭项目管理器前退出项目的激活状态。

### 2. Windows 系统启动时自动激活 WinCC 项目

当一个 WinCC 项目完成所有组态和测试正式交付现场运行时，有时为了避免没有授权的人员人为进入 Windows 操作系统进行其他操作，需要 Windows 系统重新启动后自动打开并激活该 WinCC 项目（不打开项目管理器）。Windows 系统启动时自动激活 WinCC 项目的方法如下：在 Windows 桌面单击“开始”→“所有程序”→“SIMATIC”→“WinCC”→“AutoStart”，弹出“AutoStart 组态”对话框，如图 3-10 所示。在该对话框中，通过单击“项目”中的按钮...弹出对话框来选择需要自动激活的项目，并根据需要选择激活“启动时激活项目”和“激活时允许取消”等选项后单击“添加到 AutoStart”按钮即可，最后单击“确定”按钮退出




图 3-10　“AutoStart 组态”对话框

对话框。

3. 取消激活项目

取消激活项目实际上就是退出 WinCC 运行系统，方法同样有多种：

- 可以在项目管理器的菜单栏中用鼠标单击“文件”→“√激活”。
- 可以在项目管理器的工具栏中用鼠标单击■图标。
- 可以在关闭项目管理器时选择“关闭项目并退出项目管理器”。
- 可以在过程画面中利用脚本动作组态一个按钮，在项目激活状态下单击该按钮将执行取消激活项目的任务，具体组态方法在后续的章节中介绍。

## 3.2 WinCC 的变量管理

变量系统是组态软件非常重要的组成部分。组态软件处于运行状态时，现场的运转状况将实时地反映在变量的数值中；现场操作人员监控过程数据，在计算机上发布的指令通过变量传送到现场。

WinCC 的变量系统是变量管理器（Tag Management）。WinCC 运行系统与自动化系统间的通信是依靠通信驱动程序来实现的，而自动化系统与 WinCC 项目间的数据交换是通过过程变量来完成的。在 WinCC 中，正是通过变量管理器中来组态和管理项目所需要的变量和通信驱动程序。

### 3.2.1 WinCC 变量的功能类型

WinCC 变量按照功能分类可以分为过程变量、内部变量、系统变量和脚本变量四种类型。

1. 过程变量

过程变量也可称为外部变量，前面已经提到，自动化系统（例如 PLC）与 WinCC 项目间的数据交换就是通过过程变量来完成的。WinCC 的过程变量有一个在 WinCC 项目中使用的变量名以及一个与外部自动化系统连接的数据地址，过程变量正是通过其数据地址与自动化系统进行数据通信。在 WinCC 处于运行状态时，通过自动化系统与 WinCC 项目间建立的通道连接，过程变量可以访问（读取或写入）自动化系统中与数据地址对应的数据区的过程值。因此，每一个过程变量都属于一个特定的通信驱动程序和通道单元所建立的通道连接，该过程变量也必须在其所属的通道连接的目录中创建。

过程变量创建的数目受 WinCC 系统的 Power Tags 授权数目的限制，最小是 128 个，最大是 256K 个（1K = 1024），因此，用户必须根据项目的实际需要配置相应的 Power Tags 授权数目。

2. 内部变量

内部变量不连接到外部自动化系统，因此没有对应的过程驱动程序和通道单元，也不需要建立相应的通道连接。内部变量可以用于管理 WinCC 项目中的数据或将数据传送给归档。

内部变量在变量管理器的“内部变量”目录中创建，其创建的数目不受限制。

3. 系统变量

系统变量是 WinCC 提供的一些预定义的中间变量，每个系统变量均有明确的定义，具有现成的功能，一般用来表示 WinCC 运行系统的状态。系统变量是由 WinCC 自动创建的，用户不能创建系统变量，但可以读取系统变量的值。为了与其他变量加以区分，系统变量以“@”开头，可以在整个项目的脚本和画面中使用。

4. 脚本变量

脚本变量是用户在编写 WinCC 全局脚本或画面脚本程序时定义并使用的变量，它只能在其定义时所规定的范围内使用。

### 3.2.2　WinCC 变量的数据类型

创建 WinCC 变量时，除了要给变量指定一个变量名外，还必须定义该变量的数据类型。变量的数据类型取决于用户使用的变量的数据格式。

在 WinCC 变量中，由于过程变量通过数据地址与自动化系统的相应的数据区相关联，因此过程变量的数据类型必须与自动化系统中的数据类型相匹配，主要是占用存储空间的字节数以及取值范围要匹配。如果修改现有过程变量的数据类型，则先前指定的过程变量的数据地址将被删除，需要重新输入与修改后的数据类型相匹配的数据地址。

WinCC 变量按照数据类型分，大致可以分为数值型变量、字符串型变量、文本参考型变量、原始数据类型变量和结构类型变量。

1. 数值型变量

数值型变量是组态 WinCC 项目过程中最常用的数据类型。创建数值型 WinCC 过程变量时，WinCC 数值型变量的类型可能会不同于自动化系统中所使用的数据类型，因此必须注意匹配问题。表 3-2 对比了各种数值型变量在 WinCC、STEP 7 以及 WinCC 的 C 脚本中创建时的类型声明。

表 3-2　数值型变量的 WinCC、STEP 7、和 C 脚本变量类型声明

| 变量类型名称 | WinCC 变量 | STEP 7 变量 | C 脚本变量 | 取值范围 |
|---|---|---|---|---|
| 二进制变量 | Binary Tag | BOOL | BOOL | TRUE 和 FALSE（1 和 0） |
| 有符号 8 位数 | Signed 8-bit Value | BYTE | char | -128 ~ 127 |
| 无符号 8 位数 | Unsigned 8-bit Value | BYTE | Unsigned char | 0 ~ 255 |
| 有符号 16 位数 | Signed 16-bit Value | INT | short | -32768 ~ 32767 |
| 无符号 16 位数 | Unsigned 16-bit Value | WORD | Unsigned short，WORD | 0 ~ 65535 |
| 有符号 32 位数 | Signed 32-bit Value | DINT | int | -2147483648 ~ 2147483647 |
| 无符号 32 位数 | Unsigned 32-bit Value | DWORD | Unsigned int，DWORD | 0 ~ 4294967295 |
| 32 位浮点数 | Floating-point 32-bit IEEE 754 | REAL | float | ±3.402823E+38 |
| 64 位浮点数 | Floating-point 64-bit IEEE 754 | | double | ±1.79769313486231E+308 |

2. 字符串型变量

WinCC 使用的字符串型变量根据可表示的字符集分为 8 位字符集变量和 16 位字符集变量两种。

- 8 位字符集文本变量：可以用来表示 ASCII 字符集中的字符，每个 ASCII 字符占用 1 B 的存储空间。



- 16 位字符集文本变量：一般用来表示 Unicode 字符集中的字符，每个 Unicode 字符占用 2 B 的存储空间。如需表示中文字符串，则变量类型必须为 16 位字符集文本变量。

需要注意的是，对于字符串类型的过程变量，必须指定该过程变量的长度。例如某个需要容纳 10 个字符的字符串型过程变量，若其数据类型为 8 位字符集文本变量，则其必须具有 10 个字符的长度，若为 16 位字符集文本变量，则必须具有 20 个字符的长度。

**3. 文本参考型变量**

文本参考型是 WinCC 特有的数据类型，文本参考型变量指的是 WinCC 文本库中的条目。例如，当希望交替显示不同文本块时，可创建一个文本参考型变量，将文本库中条目的相应文本 ID 分配给该变量。需要注意的是，只能将文本参考型变量组态为 WinCC 内部变量。

**4. 原始数据类型变量**

原始数据类型是 WinCC 的一种自定义数据类型，这种类型的变量多用于数据报文或用于从自动化系统传送用户数据块和将用户数据块传送到自动化系统。

外部（过程）和内部原始数据类型变量均可在 WinCC 的变量管理器中创建。原始数据类型变量的格式和长度都不是固定的，长度范围可以是 1 ~ 65 535 B，这可以由用户来定义，也可以是特定应用程序的结果。原始数据类型变量的内容也是不固定的，只有发送方和接收方能够解释原始数据类型变量的内容，WinCC 不能对其进行解释。

原始数据类型变量无法在 WinCC 的“图形编辑器”中显示，主要用于 WinCC 的以下功能模块中。

- “报警记录”：用于与具有消息的自动化系统上的消息块进行数据交换，以及消息系统的确认处理。
- “全局脚本”：在使用“Get/SetTagRaw”函数进行数据交换的脚本中。
- “变量记录”：用于过程值归档中具有过程控制变量的过程控制归档。
- “用户归档”：用于 WinCC 与自动化系统之间的作业、数据、过程确认的传送。

**5. 结构类型变量**

结构类型同样是 WinCC 提供的一种自定义数据类型，类似于 C 语言的结构体类型，是一种复合数据类型，包括多个结构元素。用户通过使用结构数据类型，将简化具有相同属性的多个对象的创建。例如，现有多台电机，每台电机具有的属性有：设定转速值、实际转速值、起动/停止状态和手动/自动控制模式，表 3-3 通过该实例对结构类型变量涉及的几个概念进行了注解。

**表 3-3　结构类型变量相关概念注解**

| 概念名称 | 定　义 | 举　例 |
|---|---|---|
| 结构类型 | 描述具有相同属性的多个对象，属性由结构元素描述和定义，至少包含一个结构元素 | 多个电机由名称为“motor”的数据类型描述 |
| 结构元素 | 结构类型的组件，每一个结构元素描述一种属性，需配置结构元素名和数据类型 | “设定转速值”：set（Signed 16-bit Value）<br>“实际转速值”：actual（Signed 16-bit Value）<br>“起动/停止”：start（Binary Tag）<br>“手动/自动”：auto（Binary Tag） |

（续）

| 概念名称 | 定　　义 | 举　　例 |
|---|---|---|
| 结构实例 | 通过结构类型创建的按照结构类型定义的对象，需指定实例名（类似于变量名） | 类型为 motor 的电机 1：motor1<br>类型为 motor 的电机 2：motor2 |
| 结构变量 | 结构变量的模板是结构元素，结构变量的名称由所使用的结构实例名称和结构元素的名称组成，中间由圆点“.”隔开 | 电机 1 的“设定转速值”：motor1. set<br>电机 2 的“设定转速值”：motor2. set<br>电机 1 的“实际转速值”：motor1. actual<br>…… |

### 3.2.3　WinCC 的通信

WinCC 的通信主要包括 WinCC 与自动化系统间的过程通信以及 WinCC 同其他应用程序之间的通信。

WinCC 与自动化系统之间的通信通过各自的过程总线来实现，例如工业以太网（Industrial Ethernet）和 PROFIBUS 等。通信由称为通道的专门通信程序来控制。作为西门子公司的组态软件，WinCC 除了提供西门子自动化系统 SIMATIC S5/S7/505（S7 不包括 S7-200 系列 PLC）的专用通道，还支持 PROFIBUS DP/FMS、DDE（Dynamic Data Exchange，动态数据交换）和 OPC 等通用通道，用于连接第三方的自动化系统。另外，WinCC 还可以以附加件（add-ons）的形式提供连接到其他控制器的通信通道。

与其他应用程序的通信，例如 Microsoft Excel、Matlab 等，借助于 OPC 接口来实现。WinCC 可以以 OPC 服务器的角色为这些应用程序提供数据，也可以以 OPC 客户端的身份访问这些应用程序的数据。

本节主要介绍 WinCC 与自动化系统之间的通信，WinCC 借助 OPC 接口与其他应用程序之间的通信请读者参考第 9 章。

#### 1. WinCC 通信的结构

WinCC 使用变量管理器集中管理项目中的变量，在运行时通过变量管理项目中的数据，例如通过过程变量采集自动化系统的数据，或通过变量归档管理存储在项目数据库中的数据。WinCC 的所有应用程序都是以 WinCC 变量的形式，通过通信驱动程序发送任务消息给自动化系统，然后自动化系统将相应的包含所请求的过程值的回复消息发送给 WinCC。这些应用程序包括 WinCC 的图形运行系统、报警记录运行系统、变量记录运行系统或全局脚本运行系统等。

WinCC 与自动化系统进行工业通信就是通过变量和过程值交换信息。WinCC 既可以读取自动化系统的过程值数据，也可以将 WinCC 中的过程变量值写回自动化系统。WinCC 与自动化系统通信的结构如图 3-11 所示。

实现 WinCC 与自动化系统之间的通信，除了在物理连接上保证正确外，还需要在 WinCC 项目中建立正确的逻辑连接。一个逻辑连接对应一个 WinCC 与自动化系统之间的通信，建立了逻辑连接，WinCC 才知道过程变量位于哪个自动化系统，以及通过什么通道来处理数据通信，自动化系统的过程值通过通道传送到 WinCC 项目的过程变量。

WinCC 与自动化系统之间的逻辑连接必须在物理连接的基础上建立。建立逻辑连接首先要在 WinCC 变量管理器中添加与自动化系统相对应的通信驱动程序，例如与西门子



自动化系统 SIMATIC S7 对应的通信驱动程序集 SIMATIC S7 Protocol Suite。而一个通信驱动程序针对不同类型的通信网络会有不同的通道单元，例如 SIMATIC S7 Protocol Suite 包含针对 MPI 网络的通道单元、针对 PROFIBUS 网络的通道单元和针对 Industrial Ethernet 网络的通道单元等等。因此，第二步要根据物理连接选择相应的通道单元，最后在该通道单元下建立 WinCC 与自动化系统间的逻辑连接。

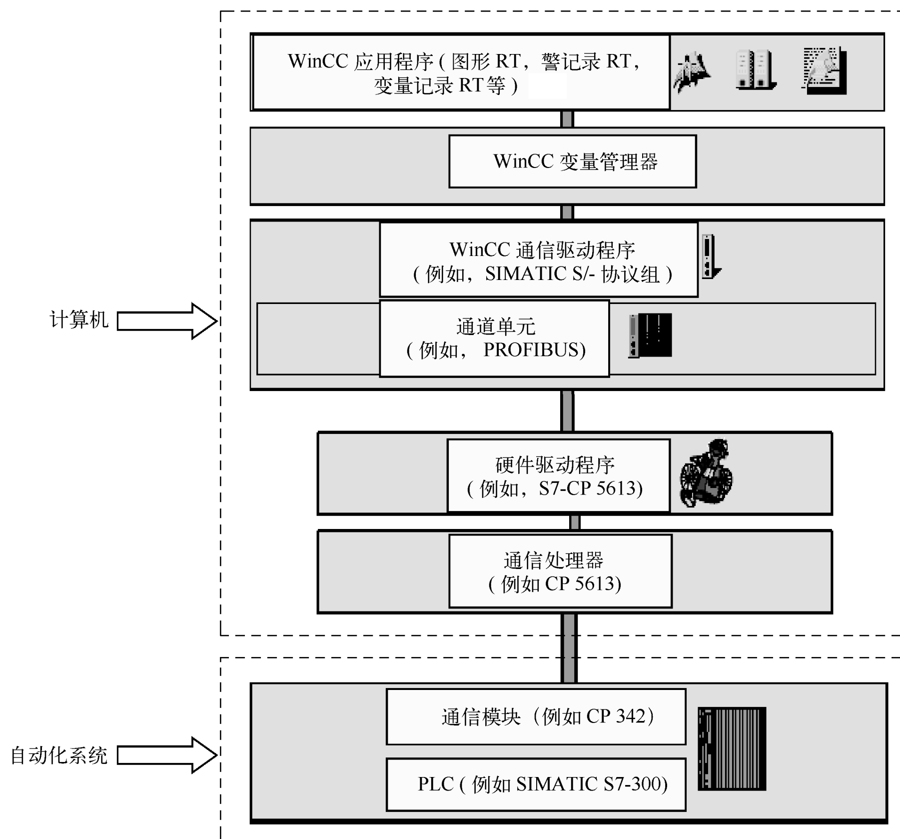



图 3-11　WinCC 通信结构图

可以看出，WinCC 与自动化系统之间的逻辑连接以分层的方式排列成三个等级，这三个等级反映在 WinCC 变量管理器的分层结构上，从上到下依次为通信驱动程序、通道单元和连接，如图 3-12 所示。

### 2. WinCC 通信的建立

以下是在 WinCC 项目中建立一个与西门子 S7 系列 PLC 的逻辑连接的详细过程：

1）添加通信驱动程序。在项目管理器的浏览窗口用鼠标右键单击“变量管理”，在弹出的菜单中选择“添加新的驱动程序”，将弹出一个“添加新的驱动程序”的对话框，在该对话框中找到“SIMATIC S7 Protocol Suite. chn”，并单击“打开”，如图 3-13 所示。

2）选择通道单元。添加新的驱动程序后，变量管理目录下新增一个“SIIMATIC S7 PROTOCOL SUITE”的子目录，如图 3-12 所示，该子目录下列出了 S7 系列 PLC 所支持的所有通道单元。根据物理连接选择相应的通道单元，例如，如果物理连接中 PLC 端通过 MPI 接口与计算机连接，则需要在通道单元列表中选择“MPI”。

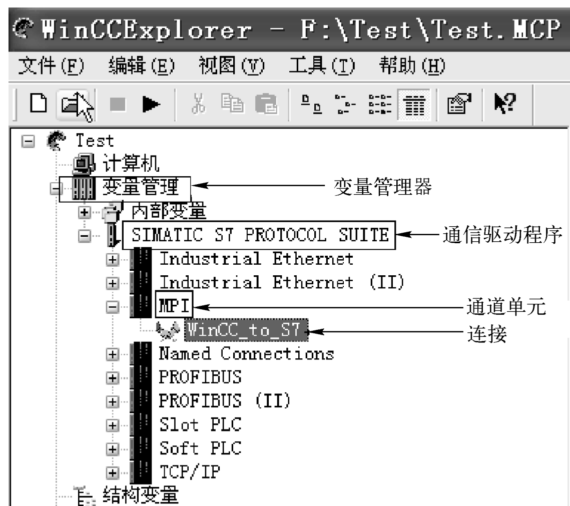



图 3-12 逻辑连接的层次结构

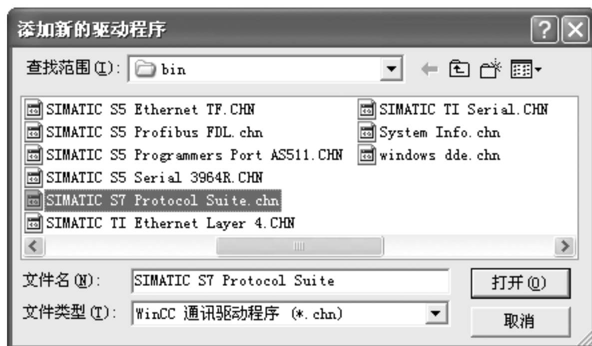



图 3-13 添加新的驱动程序

3）建立逻辑连接。用鼠标右键单击“MPI”，在弹出的菜单中选择“新驱动程序的连接”，将弹出图 3-14 所示的“连接属性”对话框。名称框中可以为新建的逻辑连接组态一个名称，服务器列表中显示了当前与 PLC 连接的计算机名称。

4）设置逻辑连接的参数。鼠标单击图 3-14“连接属性”对话框中的“属性”按钮，打开“连接参数-MPI 对话框”，如图 3-14 所示。在“S7 网络地址”框架中，需要输入“站地址”、“段 ID”、“机架号”和“插槽号”等参数，用户需要根据当前连接的 PLC 的硬件组态参数（可以在 STEP 7 中在线查看）进行设置。在图 3-14 中，“站地址”、“机架号”和“插槽号”分别对应 CPU 的 MPI 地址、机架号和插槽号。

5）设置通道单元的系统参数。用鼠标右键单击通道单元“MPI”，在弹出的菜单中选择“系统参数”，打开“系统参数”对话框，如图 3-15 所示。主要是在“单元”选项卡的“逻辑设备名称”下拉列表框中，选择 WinCC 与 PLC 通信的硬件设备，如“PC Adapter (MPI)”（运行 WinCC 的计算机，通过 RS-232 串行通信口经带适配器的通信电缆与 PLC 的 MPI 接口连接时选择该选项）或“CP5613”等。更改逻辑设备的名称后，需要重新启动 WinCC 才有效。



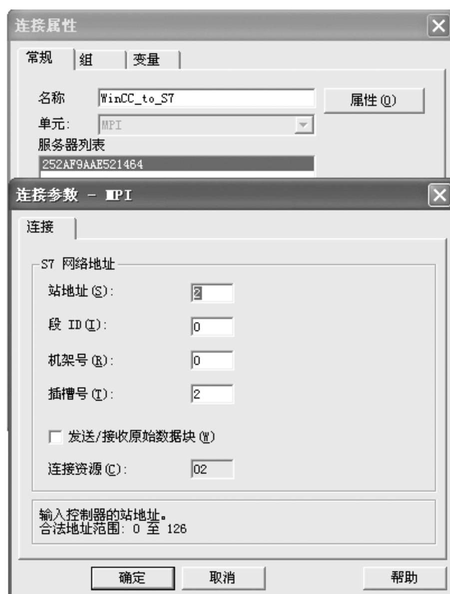



图 3-14　设置逻辑连接的属性

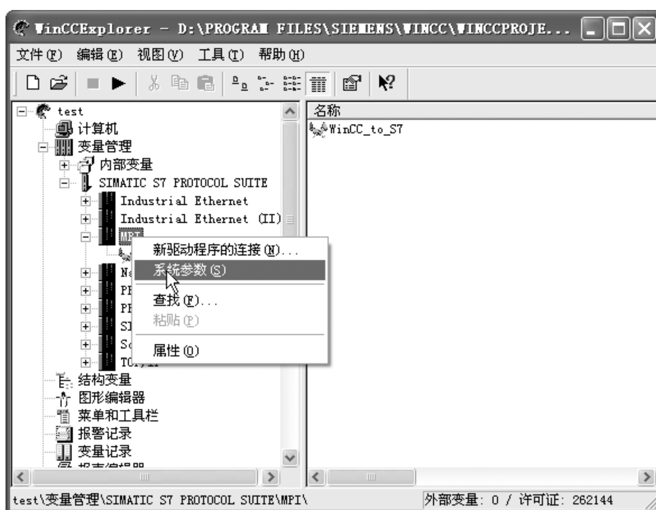



图 3-15　设置通道单元的系统参数

### 3. WinCC 通信状态的测试

在 WinCC 项目中建立一个逻辑连接后，可以对该连接的状态进行测试，以确定连接的状态是否正常。测试的条件是 WinCC 项目处于激活状态（即 WinCC 运行系统），PLC 处于正常工作状态（例如 S7-300 PLC 处于 RUN 或 RUN-P 的模式）。测试的方法如下：在项目管理器的菜单栏中单击“工具”→“驱动程序连接状态”菜单，将弹出“状态－逻辑连接”对话框，如图 3-16 所示。列表框中显示了当前项目中所有的逻辑连接及其状态，如果逻辑连接的“状态”列显示“确定”（或“OK”），表示该连接状态正常，例如名称为“WinCC_to_S7”的连接；否则连接不正常，例如名称为“NewConnnection”的连接，需要对该连接对应的硬件连接、PLC 的硬件组态参数、通道单元的系统参数和逻辑连接参数等进行检查，确定连接不正常的原因。



图 3-16　“状态 - 逻辑连接”对话框

### 3.2.4　WinCC 变量的创建

#### 1. 过程变量的创建

过程变量用于 WinCC 和自动化系统之间的通信，每一个过程变量都属于一个特定的通信驱动程序和通道单元所建立的通道连接，该过程变量也必须在其所属的通道连接的目录中创建。因此，在变量管理器中创建的过程变量，具有特定的通信驱动程序、通道单元和连接。

创建过程变量的详细步骤如下：

1）根据 3.2.3 节“WinCC 通信的建立”提供的方法，组态过程变量所属的通信驱动程序、通道单元和连接。

2）用鼠标右键单击需要创建过程变量的连接，例如图 3-12 中的“WinCC_to_S7”，在弹出的菜单中选择“新建变量”，将打开“变量属性”对话框，如图 3-17 所示。



图 3-17　“变量属性”对话框



3）在“变量属性”对话框的“常规”选项卡下的“名称”文本框中输入过程变量的名称，在“数据类型”列表框中选择过程变量的数据类型。根据需要勾选“线性标定”，还可以对过程变量进行线性标定，对 PLC 的两个过程值，给出过程变量对应的实际工程值。在“限制/报告”选项卡中，可以设置变量的上限、下限、起始值和替换值，如图 3-17 所示。

4）单击“选择”按钮，将弹出“地址属性”对话框，如图 3-18 所示，设置过程变量的地址属性。地址属性应该根据过程变量所访问的 PLC 的过程值的地址进行设定，WinCC 过程变量可以访问 PLC 的 DB 块、位内存以及输入/输出数据区。




图 3-18　过程变量的地址属性

创建后的过程变量如图 3-19 所示，由于 WinCC 软件的原因，在变量的数据窗口显示的区域符为德文，即输入类型的区域符为“E”（而不是习惯的英文“I”），输出类型的区域符为“A”。

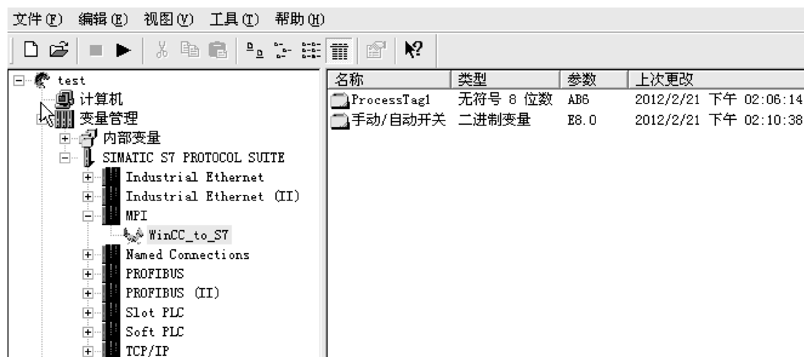



图 3-19　过程变量表

### 2. 内部变量的创建

内部变量不能连接到过程，只可以用来对项目内的数据进行管理或将数据传送给归档。创建内部变量的过程如下：

1）在项目管理器的浏览窗口打开“变量管理”目录，用鼠标右键单击“内部变量”，在弹出的菜单中选择“新建组”，打开“变量组属性”对话框，在“名称”后的文本框中输入新建变量组的名称，例如“InnerTags”，如图 3-20 所示。变量组用于在项目中创建大量变量时根据主题将其组合成变量组，例如可在项目中为每个画面创建一个变量组，这种方式使得变量的分配和检索更容易。创建过程变量时，也可以根据需要先创建变量组，然后再在组中创建变量。




图 3-20　新建变量组

2）在“内部变量”目录中用鼠标右键单击新建的变量组名称，在弹出的菜单中选择“新建变量”，打开“变量属性”对话框，如图 3-21 所示。该对话框与过程变量的“变量属性”对话框除了不需要选择变量地址以及可以选择变量更新的方式外，其他基本类似。“项目范围内更新”方式是在多用户系统的所有计算机上此变量的值相同，而“计算机本地更新”方式是在多用户系统的每台计算机上此变量的值可以不同。

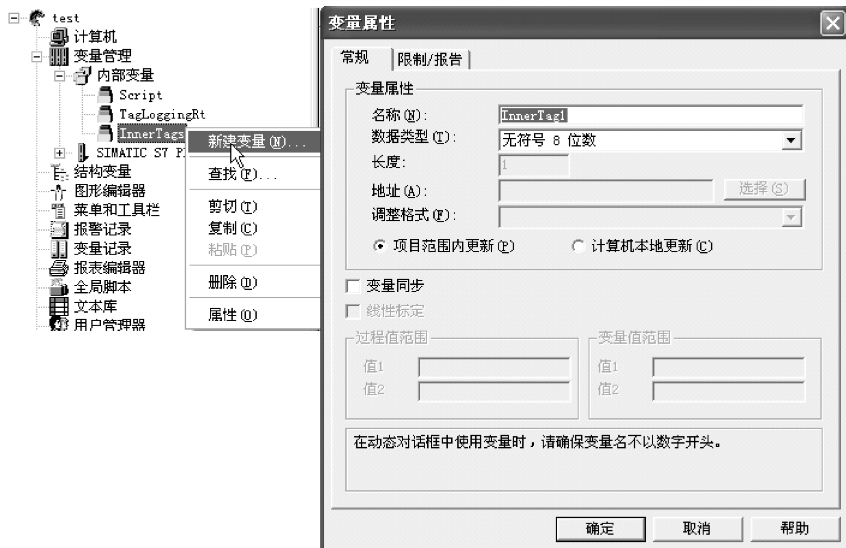



图 3-21　新建内部变量



3）单击“确定”按钮，退出对话框。

3. 结构类型变量的创建

要创建结构类型变量必须先创建相应的结构类型。当创建结构类型时，将创建不同的结构元素，创建变量时，可将所创建的结构类型分配为数据类型，从而可创建在结构类型中定义的所有结构元素所对应的变量。

下面以 3. 2. 2 节“5. 结构类型变量”中举例的结构类型“motor”为例，详细介绍创建结构类型变量的过程。

（1）首先创建结构类型“motor”

创建结构类型需在项目管理器的“结构变量”下进行，具体步骤如下：

1）用鼠标右键单击项目管理器浏览窗口的“结构变量”，在弹出的菜单中选择“新建结构类型”，打开“结构属性”对话框，如图 3-22 所示。列表框中“NewStructure”为新建的结构类型名称，用户可以鼠标右键单击该名称，并在弹出的菜单中选择“重命名”为其分配一个新的名称，例如“motor”。

2）单击“新建元素”按钮，可以为新建的结构类型添加“结构元素”，默认名称为“NewTag”，数据类型为“short”，用鼠标右键单击新建的结构元素，在弹出的菜单中可以修改结构元素名称和数据类型，如图 3-23 所示。选中已经添加的结构元素，然后单击“删除元素”按钮，可以删除该结构元素。

注意：结构类型中的每一个结构元素都可以选择是过程变量（对话框中称为外部变量）还是内部变量。如果设置为过程变量，则需要设置在“AS”段中的偏移量，如图 3-23 右图的黑色矩形框内，该偏移量确定以字节为单位的结构元素离起始地址的距离，新建或复制的类型为过程变量的元素的偏移量将自动增加，用户请根据需要调整设置。

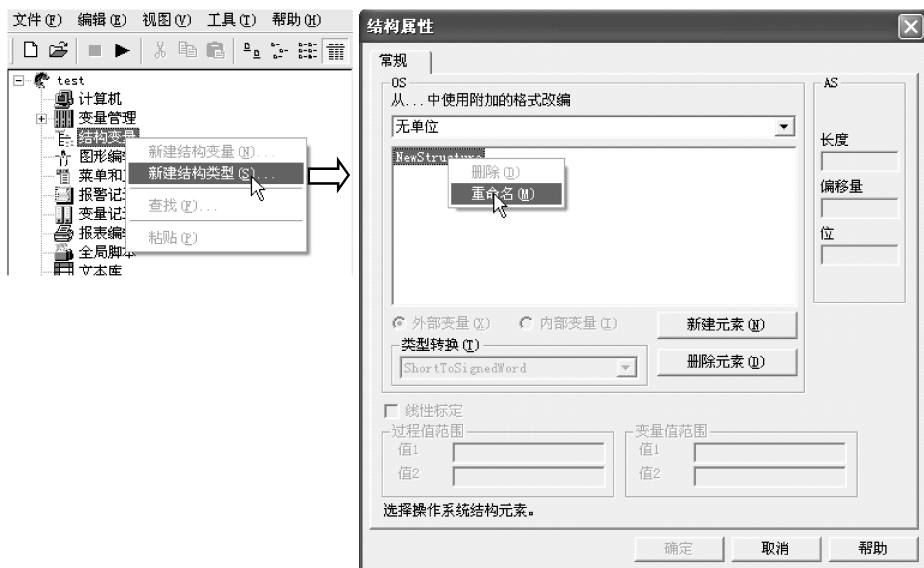



图 3-22　新建结构类型

3）所有结构元素添加并编辑后，单击“确定”按钮退出对话框。新建的结构类型“motor”将出现在项目管理器浏览窗口的“结构变量”目录下。

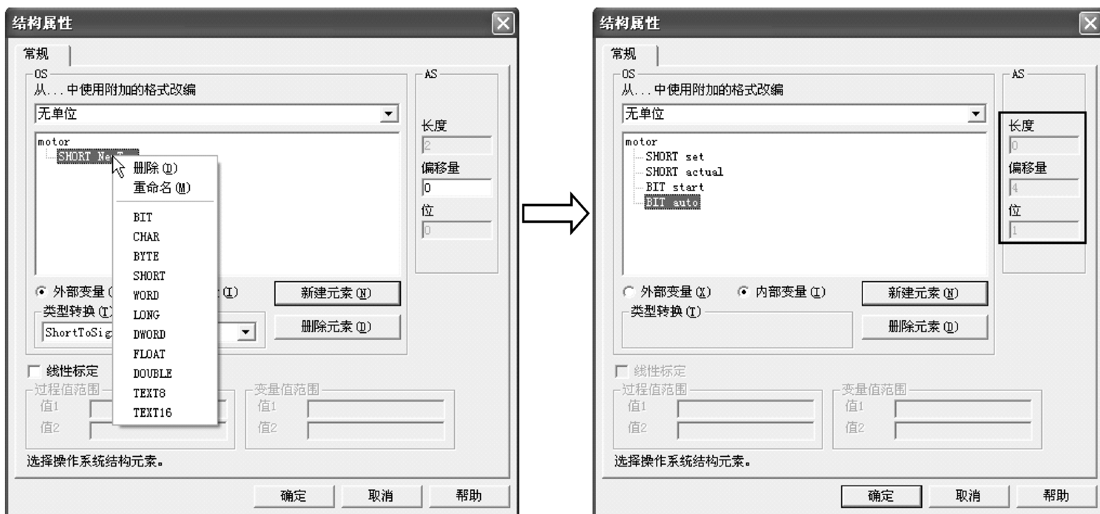



图 3-23　创建结构类型的结构元素

（2）创建结构类型变量

创建结构数据类型后，就可以创建结构类型变量了。由于结构类型中包含有多个元素，一个结构元素对应的是一个结构变量，因此会同时生成多个结构变量，一般称该过程为创建一个结构实例（结构实例的概念请参考 3. 2. 2 节的表 3-3），一个结构实例是由多个结构变量组成的。创建结构实例的过程和创建单个过程变量或内部变量的过程类似，只是在选择数据类型时选择相应的结构类型就可以了。如图 3-24 所示。




图 3-24　创建结构类型实例

注意：

- 如果所创建的结构类型中包含类型为过程变量的结构元素，则该结构类型对应的结



构实例必须在相应的逻辑连接目录下（如图 3-12 的“WinCC_to_S7”）创建，“内部变量”目录下无法创建。该实例创建后，实例中包含的类型为过程变量的结构变量存放在逻辑连接目录下，类型为内部变量的结构变量在“内部变量”目录下显示，如图 3-25 所示。

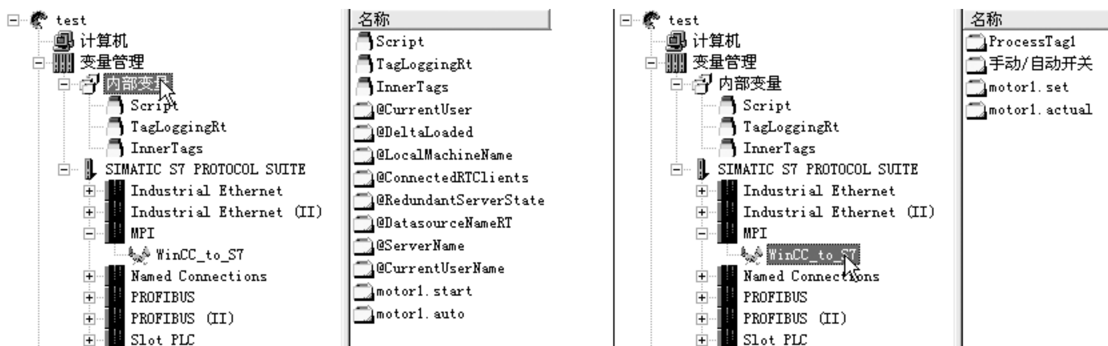



图 3-25　结构类型变量表

- 在应用结构类型创建结构实例之前应该完成所有的设置，随后只可修改所创建的结构类型变量的属性，如线性标定、地址和限制/报告等内容。如果要修改结构类型的属性，必须首先删除所有相关联的结构变量。
- 结构类型变量创建后，无法进行单个的删除，必须在其对应的结构类型目录下删除该结构类型变量所属的结构实例，该结构实例所包含的结构类型变量都将被删除。

## 3.3　组态技巧与技术问答

本节将介绍一些组态 WinCC 项目和变量的技巧，并针对项目和变量组态过程中遇到的一些常见的技术问题进行分析和解答。

### 3.3.1　移植（升级）WinCC 项目的方法

从 WinCC V6.0 开始，在数据组织方面与以前的版本有着显著的不同。例如，从 WinCC V6.0 开始，使用 Microsoft 的数据库平台 Microsoft SQL Server（WinCC V6.0 使用 SQL Server 2000，V6.2 使用 SQL Server 2005）来管理运行系统数据和组态数据，而不再使用 Sybase SQL Anywhere。为了使在 WinCC V5.0 SP2 或 WinCC V5.1 中创建的项目在 WinCC V6.0 及以上版本也能工作，项目数据必须首先通过移植做相应的调整。WinCC V6.0 及以上版本均提供了一个项目移植器，用于自动移植 WinCC 项目的组态数据、运行系统数据和归档数据。

在移植项目之前，建议用户将原版本的项目进行备份。移植项目到 WinCC V6.0 或 V6.2 的流程图如图 3-26 所示。项目移植器是 WinCC 的一个工具，单击 Windows 菜单“开始”→“所有程序”→“SIMATIC”→“WinCC”→“Tools”→“Project Migrator”，打开“WinCC 项目移植器”对话框，接下来按照对话框中的提示操作即可，如图 3-27 所示。

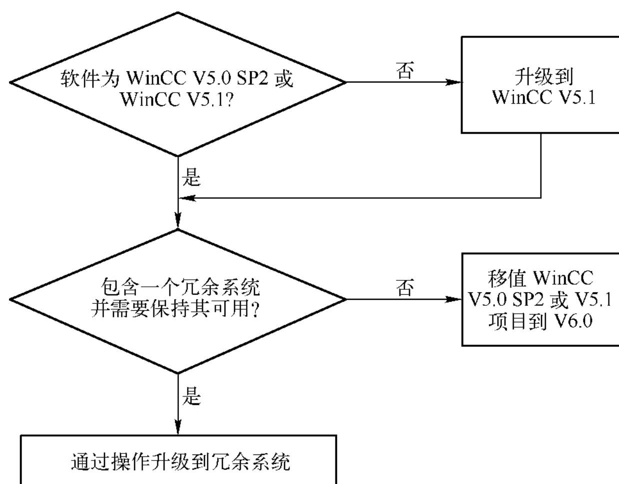



图 3-26　移植项目到 WinCC V6.0 的流程图

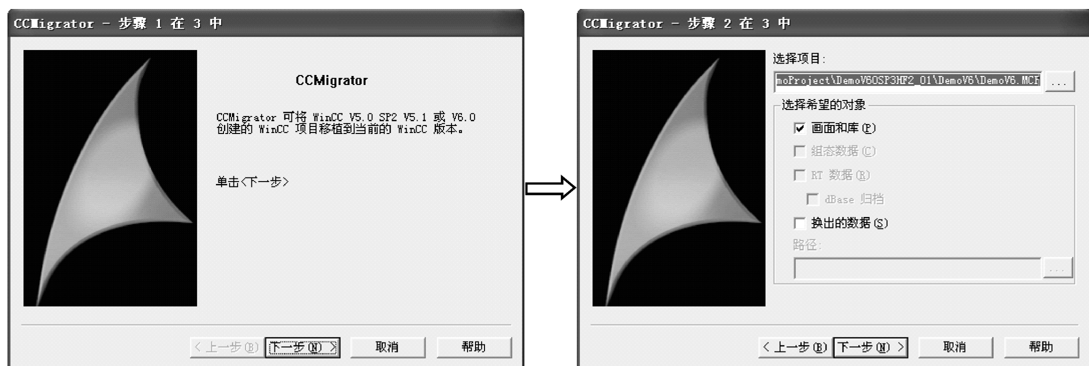



图 3-27　项目移植器对话框

如果必须访问项目先前版本的归档数据，则必须将归档移植到 WinCC V6.0，使用项目移植器移植归档数据和 dBASE III 归档。

为了使利用 WinCC V5.0 SP2 或 V5.1 所创建的多用户项目在 WinCC V6.0 及以上版本中能够正常工作，则需要移植系统中所有服务器上的单个多用户项目。如果原项目使用了多客户机，则分别单独移植多客户机的项目数据。

对于正常操作中的冗余系统，不用取消项目的激活状态就可以在冗余系统中对项目进行升级。此时，WinCC 将按规定的次序升级服务器、客户机和多客户机。

对于早于 WinCC V5.0 SP2 版本的 WinCC 所创建的项目，必须先将 WinCC 升级到 V5.1，并在 WinCC V5.1 中移植项目，再安装 WinCC V6.0，并使用项目移植器移植项目。

注意：

- 移植 WinCC V5.0 SP2 或 V5.1 的项目时，除了需要 WinCC V6.0 或 V6.2 以及 SQL Server，还需要安装 Sybase SQL Anywhere 7.0，用于将项目的数据转换到 SQL Server，项目移植后才可以卸载 Sybase SQL Anywhere 7.0。Sybase SQL Anywhere 7.0 软件包在 WinCC V6.0 或 V6.2 的安装光盘中
- WinCC V5 项目不能直接移植到 WinCC V7，首先需要将 WinCC V5 的项目移植到 WinCC V6，然后移植 WinCC V6 的项目到 WinCC V7。



## 3.3.2 复制 WinCC 项目的方法

复制项目是将项目和所有重要的组态数据复制到同一台计算机的另一个文件夹或网络中的另一台计算机。建议用户在组态 WinCC 项目的过程中定期对项目进行备份。

由本章 3.1.4 小节可知，WinCC 项目以及所有重要的组态数据都存放在与项目名称相同的项目文件夹中，因此，用户首先想到的是用 Windows 的复制粘贴功能来复制 WinCC 项目。实际上，WinCC 提供了专门的工具——项目复制器（Project Duplicator）来复制项目，建议用户优先选用项目复制器备份 WinCC 项目，因为通过项目复制器复制后的项目名称可以与原项目的名称不同，而且不能使用 Windows 的复制粘贴功能复制冗余服务器项目。使用项目复制器，只复制项目和项目的所有组态数据，不复制运行系统数据。

在 Windows 的“开始”菜单中选择“所有程序”→“SIMATIC”→“WinCC”→“Tools”→“Project Duplicator”，打开“WinCC 项目复制器”对话框，如图 3-28 所示。在“选择要复制的原项目”框中输入希望复制的项目路径及项目主文件，或单击“[...]”按钮浏览选择需要的项目。单击“另存为”按钮，打开“另存为 WinCC 项目对话框”对话框，选择复制项目的存储文件夹，并在“文件名”框中输入项目名称，该名称既可以与原项目名称相同，也可以不相同，单击“保存”按钮即可。

冗余服务器上的 WinCC 项目必须与主服务器项目完全相同。如果组态了一套冗余系统，则每当主服务器项目进行了修改，必须对冗余服务器上的项目进行同步更新。前面已经提到，复制冗余服务器项目，不能使用 Windows 的复制粘贴功能，只能通过 WinCC 项目复制器。在图 3-28 中，分别选择原项目和目的项目的存储位置，单击“复制”按钮，就可以开始复制冗余服务器的项目。

注意：如果将 WinCC 项目复制到另一台计算机，由于原计算机名称仍保留在项目的计算机属性中，必须将项目计算机属性中的计算机名称更改为当前的计算机名称才能编辑和运行该项目。

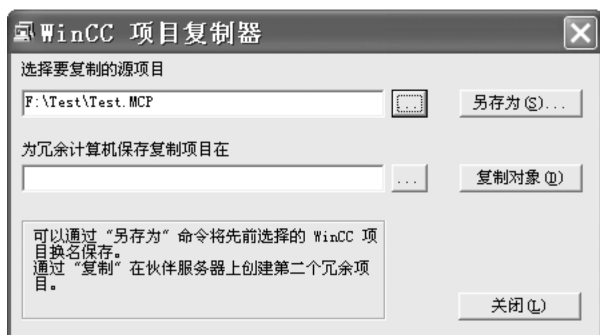



图 3-28 “WinCC 项目复制器”对话框

## 3.3.3 WinCC 项目无法正常激活的解决办法

### 1. 问题现象

按本章 3.1.5 小节介绍的激活 WinCC 项目方法对项目进行激活后，“激活”对话框中

的进度条始终保持图 3-29 所示的状态不变化（根据项目不同，进度条的百分比状态可能存在差异），无法进入运行系统，无法进行任何组态操作。特别是在项目的计算机属性中激活了“变量记录运行系统”、“报警记录运行系统”等功能后，激活项目时这种现象出现的几率非常高。

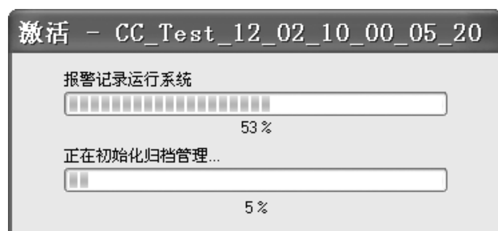



图 3-29　WinCC 项目无法激活的现象

**2. 问题原因**

当前运行的 WinCC 项目的项目名称或存放路径中存在中文字符，WinCC 软件对中文的兼容性并不好，如果 WinCC 项目的项目名称或存放路径中存在中文字符，将严重影响该项目的正常运行（激活）。例如“D:\WinCC 项目\test. mcp”和“D:\WinCCProject\我的项目 . mcp”这样的项目路径或项目名称都是不允许的（计算机驱动器的卷标可以是中文）。

**3. 解决办法**

首先执行“reset_wincc. vbs”脚本程序终止 WinCC 项目激活的停滞状态并退出 WinCC 项目管理器（详细方法请参阅 3.3.4 小节）。建议用户在组态 WinCC 项目时，将新建或复制的 WinCC 项目存放在全英文路径中，如果当前 WinCC 项目的项目名称存在中文字符，可以按本章 3.3.2 小节所介绍的，用 WinCC 项目复制器复制该项目，并将复制后的项目名称改为不含中文的名称。

## 3.3.4　WinCC 出现“死机”现象的解决办法

**1. 问题现象**

在激活和取消激活 WinCC 项目时，出现类似于 3.3.3 小节图 3-27 所示的状况，激活项目或取消激活项目的进度条一直保持一个固定的百分比不变化，WinCC 组态系统无法进行任何操作，即 WinCC 出现“死机”。

**2. 问题原因**

出现这种问题的原因是多方面的，但最根本原因的有两个：

- WinCC 项目在激活和取消激活时以及 WinCC 运行系统占用较多的计算机系统资源，例如系统的内存和 CPU 进程。
- 本章 3.3.3 小节所分析的 WinCC 软件对中文的兼容性不够好。

**3. 解决办法**

用户首先想到的是采用 Windows 的〈Ctrl + Alt + Delete〉组合键打开“Windows 任务管理器”对话框来中断“死机”状态。实际上，WinCC 提供了一个脚本程序“Reset_WinCC. vbs”（不区分大小写），执行该脚本程序不仅可以中断 WinCC 的“死机”状态，



而且可以完全退出 WinCC 系统，包括与 WinCC 有关的所有进程。该脚本程序可以采用两种方法执行：

- 选择 Windows“开始”菜单中的“运行”项，打开“运行”对话框，在“打开”后的命令框中输入“Reset_WinCC. vbs”，并单击“确定”按钮执行脚本程序。
- 双击“C:\Program Files\Siemens\WinCC\bin\Reset_WinCC. vbs”（这里假定 WinCC 软件安装在计算机硬盘驱动器的（C:）区运行脚本程序。

实际上，如何避免 WinCC“死机”现象的出现，才是问题的关键。建议用户规范地使用 WinCC，例如，在 WinCC 激活或取消激活的过程中尽量不做其他的 Windows 操作，尽量在退出 WinCC 运行系统的状态下进行 WinCC 的组态等。

注意：在使用 Windows 的复制粘贴功能复制 WinCC 项目或在删除 WinCC 项目文件夹中的文件等操作时，经常会遇到“…文件正在使用”的问题，用户可以用“Reset_WinCC. vbs”脚本程序关闭与 WinCC 有关的所有进程，然后再执行这些操作。

### 3.3.5 WinCC 项目中直接调用 STEP 7 变量的方法

WinCC 项目通过过程变量实现访问自动化系统的过程值，对于一个数据通信量非常大的系统来说，建立变量是一件十分重要而繁琐的工作。在自动化系统中，例如 S7－300/400 PLC，需要在其组态软件 STEP 7 中建立变量表、DB 块等，用于保存采集的过程值。WinCC 项目要访问这些过程值，需要在项目中建立过程变量来访问其对应的变量表或 DB 块。如果采用本章 3.2.4 小节介绍的方法创建过程变量，工程人员相当于重复了在 STEP 7 中建立变量表和 DB 块等的工作。有没有一种方法能实现在 WinCC 项目中直接调用在 STEP 7 中已经建立的变量呢？答案是肯定的，西门子一直倡导 TIA（全集成自动化）的理念，实现了统一的平台、统一的数据和统一的通信的目标。

#### 1. 在 WinCC 项目中直接调用 STEP 7 的变量的意义

在 WinCC 项目中直接调用 STEP 7 的变量，可以将建立变量的工作量减少一半，同时将建立变量的出错概率也减少一半，从而减少了相应的排错工作，大大提高了工作效率，加快了工程进度。

#### 2. 在 WinCC 项目中直接调用 STEP 7 的变量的前提条件

在 WinCC 项目中直接调用 STEP 7 的变量的前提条件是 WinCC 的项目必须集成在 STEP 7 项目中。而要使用 WinCC 与 STEP 7 的集成功能，必须满足以下条件：

- 计算机上必须同时安装了 WinCC 和 STEP 7。
- 必须在安装 WinCC 之前安装 STEP 7。
- STEP 7 与 WinCC 的版本必须兼容。可以查阅 WinCC 与 STEP 7 的版本兼容性列表（相关内容请参阅第 2 章的 2.2 节）
- 安装 WinCC 时，必须安装与集成功能相关的组件。因此，在安装 WinCC 时，选择“自定义安装”，并在“选择组件”对话框中选中方框内所包含的所有组件，如图 3-30 所示。

#### 3. 在 Simatic Manager（STEP 7）中集成 WinCC 项目

WinCC 项目集成在 STEP 7 项目中，可以是在 STEP 7 项目中创建新的 WinCC 项目，也可以把现成的 WinCC 项目集成到 STEP 7 项目中。




图 3-30　安装与集成功能相关的 WinCC 组件

（1）在 STEP 7 项目中创建新的 WinCC 项目

在创建新的 STEP 7 项目后，可以在 SIMATIC Manager 里直接组态新的 WinCC 项目。方法是：在 SIMATIC Manager 里，右键单击 STEP 7 项目的名称，在弹出的菜单中选择“插入新对象”→“OS”，如图 3-31 所示。此时系统会在 STEP 7 项目文件夹里建立一个 WinCC 项目文件夹。用户可以在 OS 站上单击右键，在弹出的菜单中选择“重命名”来修改 WinCC 项目的名称。可以通过在名称上单击右键，选择“打开对象”打开并编辑该 WinCC 项目，如图 3-32 所示。

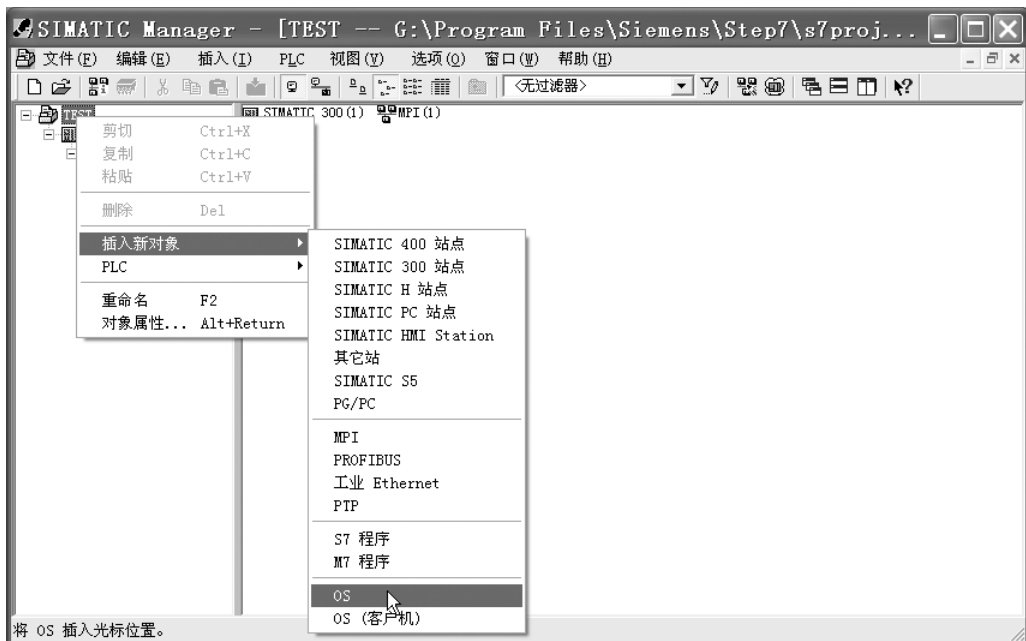



图 3-31　在 STEP 7 项目中插入 OS 站（新建 WinCC 项目）



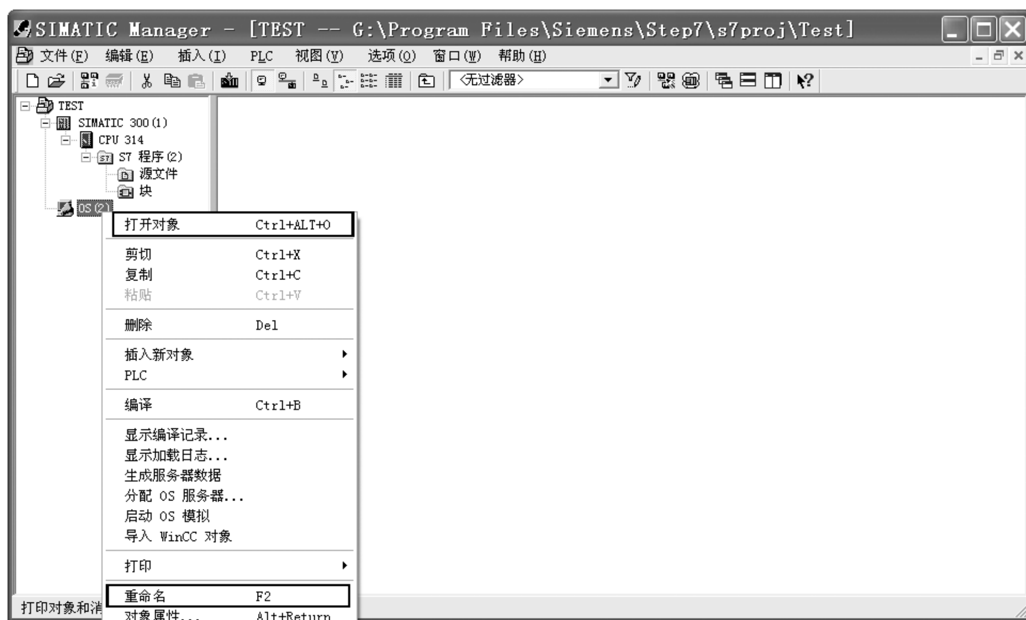



图 3-32 修改 WinCC 项目名称和打开 WinCC 项目

（2）将现成的 WinCC 项目集成到 STEP 7 项目中

按如下步骤将现成的 WinCC 项目集成到 STEP 7 项目中：

1）按图 3-31 的方法在 STEP 7 项目中插入一个 OS 站，然后把 OS 站的名称改为已有的 WinCC 项目名称。

2）在 STEP 7 项目文件夹中删除插入 OS 站后产生的 WinCC 项目文件夹，其位置在 STEP 7 项目文件夹的“wincproj”文件夹下。例如图 3-31 所示的 STEP 7 项目在“G:\Program Files\Siemens\Step7\S7Proj”文件夹中，则插入 OS 站后产生的 WinCC 项目文件夹位于“G:\Program Files\Siemens\Step7\S7Proj\Test\wincproj\”。

3）最后参考本章 3.3.2 小节介绍的方法用项目复制器将已有的 WinCC 项目复制到 STEP 7 项目的“wincproj”文件夹下。

**4. 将变量从 STEP 7 项目传送到 WinCC 项目**

WinCC 项目集成到 STEP 7 项目后，可以将 STEP 7 项目的符号表变量和 DB 块变量一次性地传入到集成的 WinCC 项目中，该 WinCC 项目可以直接使用这些符号表变量和 DB 块变量。需要注意以下几点：

- WinCC 中变量的属性（如：限值、初始值等）都可以在 STEP 7 中进行设置。
- 传送完成后，经过设置的符号表和 DB 块变量自动出现在相应的逻辑连接中。
- 只能传送基本数据类型的变量，不能传送字符串、数组、Date and Time 等复杂数据类型。
- 为了保持数据的一致性，传送过后的变量不能在 WinCC 项目中直接修改或删除，必须在 STEP 7 项目中进行修改并再次传送。

以下介绍将变量从 STEP 7 项目传送到 WinCC 项目的详细过程。

（1）为需要传输的变量标记“传送标志”

在传送前，要为需要传送的符号表和 DB 块变量做“传送标志”。两种变量标记“传送标志”的方法存在差异。

- 符号表变量标记“传送标志”。如图 3-33 所示，在需要传送的变量上单击右键，在弹出的菜单中选择“特殊对象属性”→“操作员监控”，打开“操作员监控”对话框，激活复选框“操作员监控”（图 3-33 矩形框内），并单击“确定”按钮关闭对话框，此时该变量前将显示绿色小旗，就如图 3-31 中符号表变量“IntTag”一样。

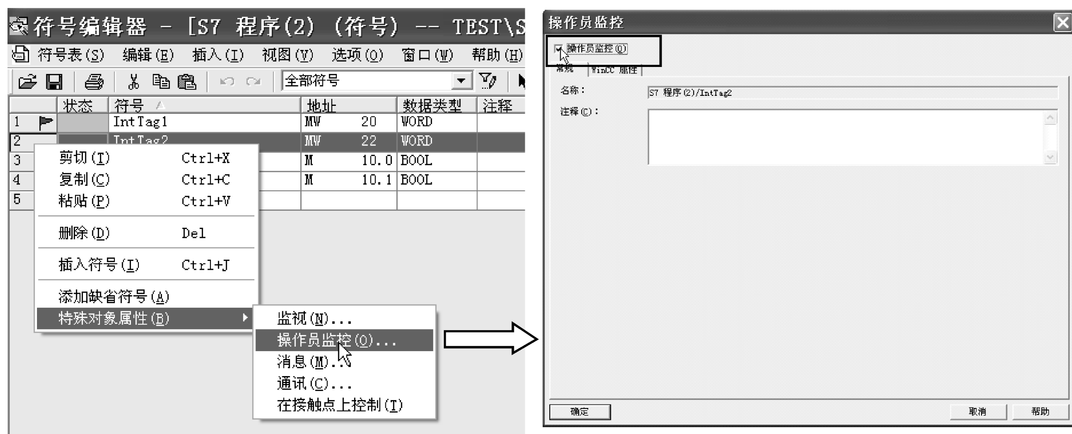


图 3-33　为符号表变量标记“传送标志”

如需对符号表中成批变量进行标记操作，可以使用“〈Shift〉+ 鼠标左键”进行批选择，或者使用“〈Ctrl〉+ 鼠标左键”对成批选择的变量进行个别增减。

- DB 块变量标记“传送标志”。如图 3-34 所示，在需要传输的 DB 块变量上单击右键，在弹出的菜单中选择“对象属性”，打开“属性 - 参数”对话框，在对话框的第一行输入图中所示参数，并单击“确定”按钮退出对话框，此时被选中变量前被用红色小旗标志，就如图中“DB_VAR2”变量一样。

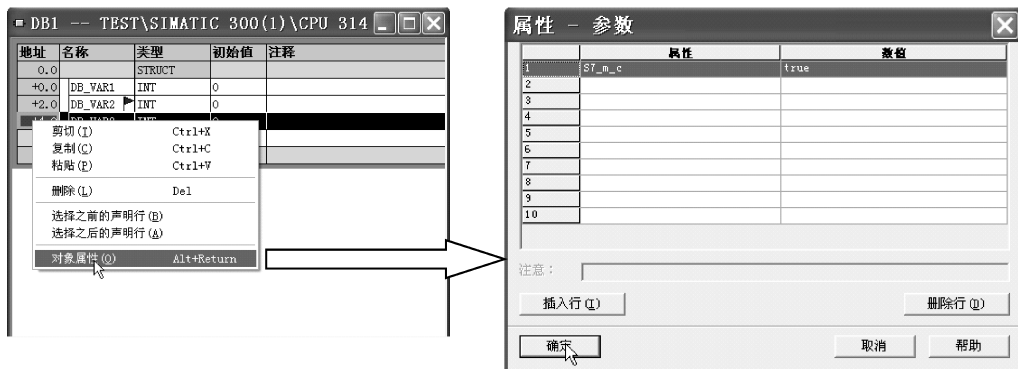


图 3-34　为 DB 块变量标记“传送标志

DB 块里的变量被标记“传送标志”后，必须激活 DB 块的“操作员监控”功能才能实现 DB 块变量的传送，如图 3-35 所示。



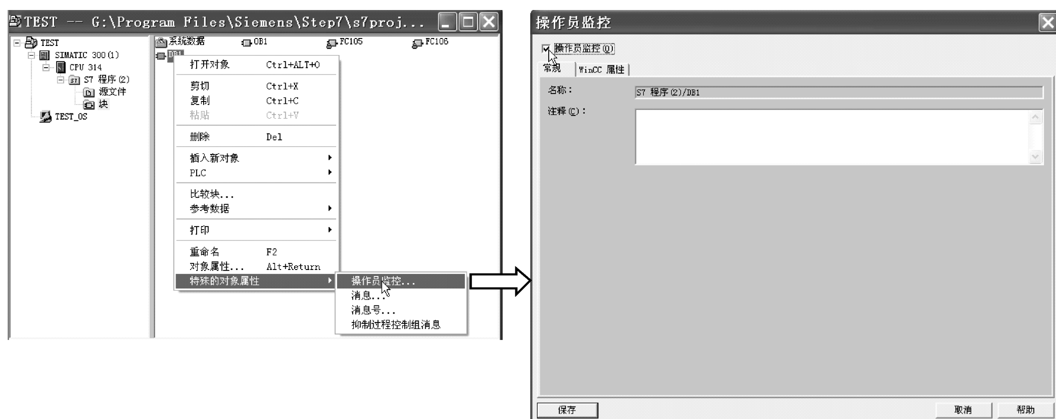


图 3-35　激活 DB 块“操作员监控”功能

（2）将被标志的变量传送到 WinCC 项目文件中

将被标志的变量传送到 WinCC 项目中，需要对 WinCC 项目对应的集成在 STEP 7 中的 OS 站进行编译，编译的详细步骤如下：

1）在 SIMATIC Manager 中用鼠标右键单击 OS 站名称，在弹出的菜单中选择“编译”，打开编译 OS 站的向导对话框，如图 3-36 所示。

2）单击图 3-36 右图所示对话框的“下一步”按钮，弹出编译向导的下一个对话框，如图 3-37 左图所示。在该对话框中，如果项目中已经建立多个 PLC 站、OS 站或多个连接，需要在该对话框中选择被编译的站和连接。选择完成后，单击“下一步”按钮。

3）在接下来的对话框中，用户需要选择编译的（向 WinCC 传送的）数据种类和范围，如图 3-37 右图所示。实现 WinCC 直接调用 STEP 7 变量，需要选择“Tags and message”（变量和消息）选项，而范围可以是整个项目的所有相关数据（“Entire OS”），也可以只对进行过更改的数据（“Changes”）进行编译和传送。选择完成后，单击“下一步”按钮。

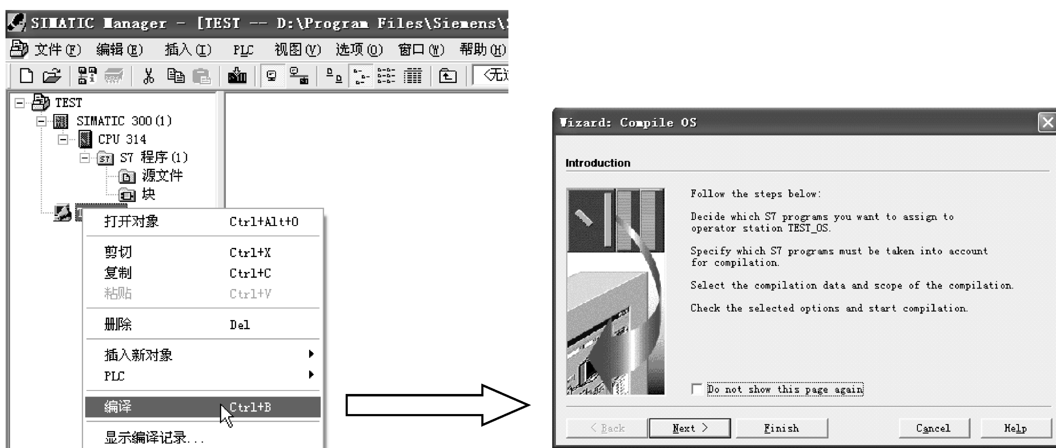


图 3-36　编译集成的 WinCC

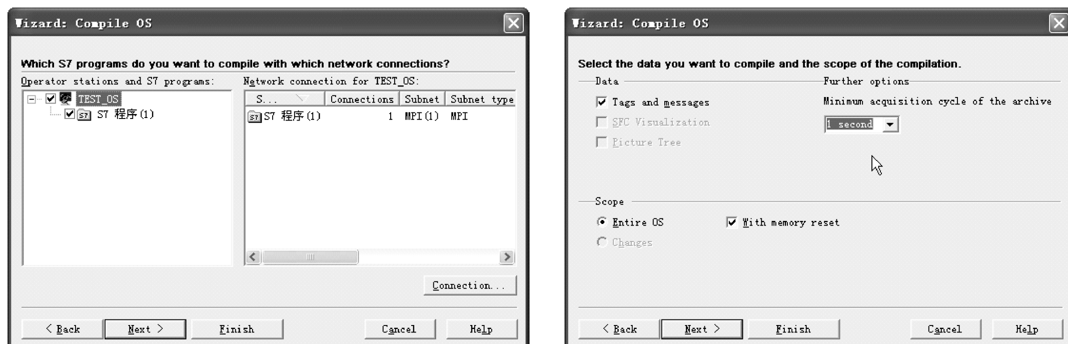



图 3-37　编译 OS 站向导

4）在接下来的对话框中单击“Compile”，编译过程开始，将弹出图 3-38 左图所示的编译进度对话框，编译过程中还会弹出右图所示“选择消息号分配”对话框，单击“确定”按钮，继续编译过程，直至结束。

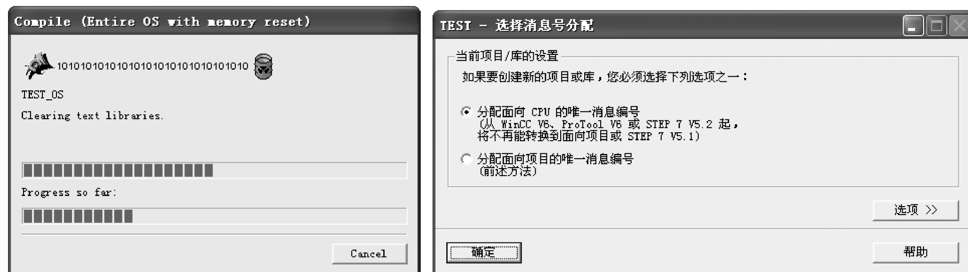



图 3-38　编译进度对话框和选择消息号分配对话框

编译完成后系统会提示编译是否成功，如果失败会弹出相应的记录文件。编译成功后，打开 WinCC 项目（在图 3-36 左图的弹出菜单中选择“打开对象”打开 WinCC 项目）。可以看到，系统已经在变量管理器里自动生成了相应的 WinCC 变量，如图 3-39 所示。

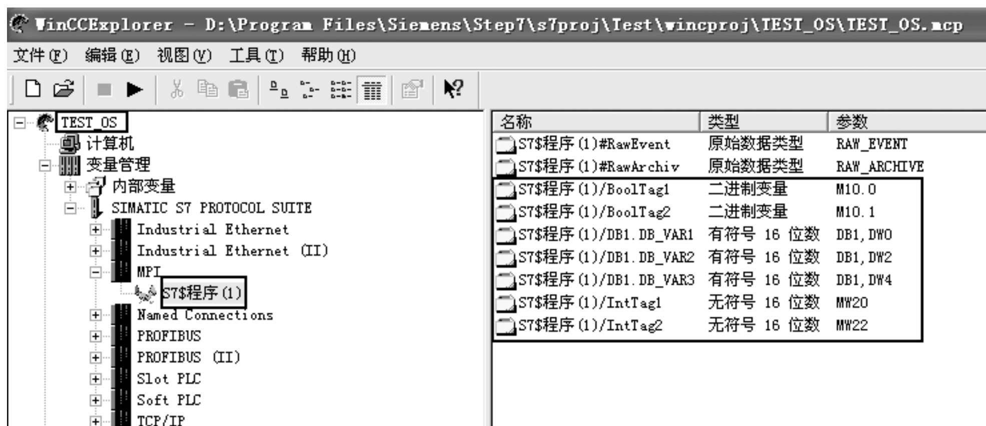



图 3-39　传送到 WinCC 项目的 STEP 7 变量



5. WinCC 项目中直接调用 STEP 7 变量

在 WinCC 项目中调用 STEP 7 变量和调用 WinCC 项目中创建的变量的方法相同。在 WinCC 项目需要调用变量的位置进行变量选择，出现“变量选择”对话框后，选中对话框右上角的“STEP 7 Symbol Server”复选框。此时，对话框左侧变量选择列表中将显示“STEP 7 Symbol Server”列表。从改列表中可以显示 STEP 7 符号表和 DB 块里的所有变量，如图 3-40 所示。

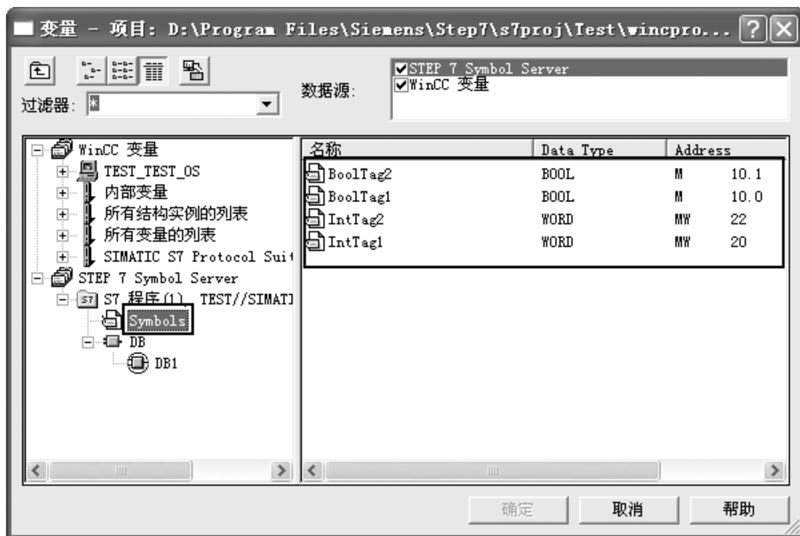



图 3-40　WinCC 中调用 STEP 7 变量

## 3.3.6　WinCC 变量的导入/导出方法

变量的导入/导出功能可以比较高效地处理大批量的变量，WinCC 可以通过两种方法来实现变量的导入/导出功能：一种方法是通过“WinCC Smart tools”（WinCC 智能工具）的变量导入/导出工具实现，另一种方法是使用“WinCC Configuration Tool”（WinCC 组态工具）在 Microsoft Excel 中导入/导出变量。

1. 使用“WinCC Smart tools”智能工具导入/导出变量

要使用“WinCC Smart tools”智能工具导入/导出变量，首先必须安装“WinCC Smart tools”，在 WinCC 安装光盘的目录“\WinCC\setup\Products\SC_SMARTTOOLS”中双击“setup. exe”启动安装过程，在“Select Components”对话框中勾选“VarExpImp”组件，如图 3-41 所示。

“变量导入/导出”智能工具是以 WinCC API 为基础的独立的应用程序，可以用来将项目的所有 WinCC 变量导出到 ASCII 文件，再将其导入到另一个项目。导出过程中，将生产三个文件：“项目名称_cex. csv”文件描述了项目中的逻辑连接；“项目名称_dex. csv”文件描述了项目中的结构数据类型；“项目名称_vex. csv”描述了项目中的所有 WinCC 变量。

单击 Windows“开始”菜单，并选择“所有程序”→“SIMATIC”→“WinCC”→“Tools”→“TAG EXPORT IMPORT”，可以启动“变量导入/导出”工具，如图 3-42 所示。

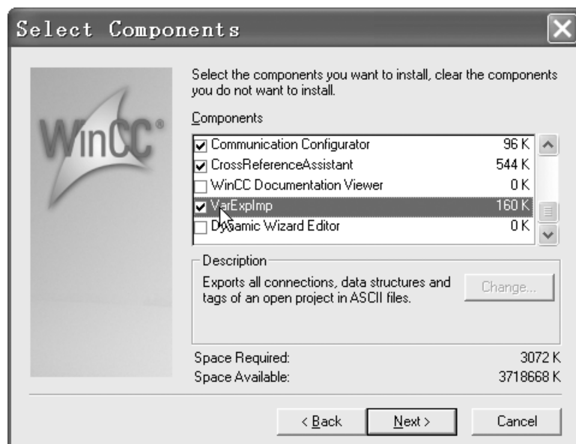



图 3-41　选择安装“VarExpImp”组件

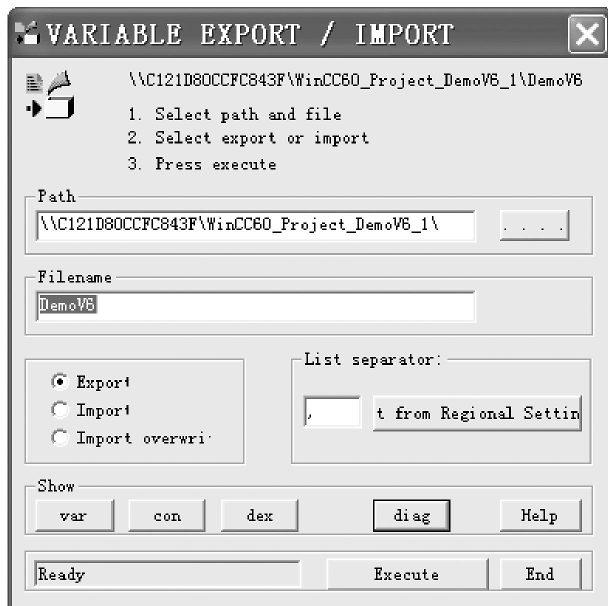



图 3-42　变量导入/导出工具

（1）变量导出

变量导出的详细步骤如下：

1）首先启动 WinCC 并打开想要从中导出变量的项目，启动“变量导入/导出”工具。

2）选择想要导出到其中的文件的路径和名称。开头仅需不具有扩展名的文件名称。

3）将模式设置为“Export（导出）”。

4）按下“Execute（执行）”按钮，确认消息框中的条目。

5）一直等到状态栏中显示“End Export/Import（结束导出/导入）”。

6）通过单击“show”中相应的按钮“var”（变量）、“con”（连接）、“dex”（结构）和“diag”（记录册），可以查看导出后生成的文件。

注意：空组不导出，下划线“_”为命名保留，因此选择导出文件名称时决不能包含下划线。



（2）变量导入

变量导入的步骤如下：

1）首先启动 WinCC 并打开想要导入变量到其中的项目。

2）将要导入连接到其中的所有通道驱动程序必须在项目中都可用。如果需要，将缺少的驱动程序添加到项目中。

3）启动“变量导入/导出”工具。

4）选择想要从中导入的文件的路径和名称。开头仅需不具有扩展名的文件名称。使用选择对话框时，单击三个导出文件中的一个。

5）选择“Import（导入）”或“ImportOverwrite（导入重写）”模式。在“导入重写”模式中，目标项目中已存在的变量使用相同名称的导入变量进行重写。在“导入”模式中，一条消息将写到日志文件中，目标项目中的变量保持不变。

6）按下“执行”按钮，确认消息框中的信息。

7）一直等到状态栏中显示“结束导出/导入”。

8）在 WinCC 变量管理器中查看生成的数据。

注意：在使用“变量导入/导出”智能工具导入或导出变量时，WinCC 项目必须处于打开并取消激活的状态。

**2. 使用“WinCC Configuration Tool”在 Microsoft Excel 中导入/导出变量**

在 WinCC 安装光盘中，可以选择安装“WinCC Configuration Tool”，安装此选件后，在 Microsoft Excel 中可以打开 WinCC 项目中的变量列表。“WinCC Configuration Tool”位于 WinCC 安装光盘的“\WinCC\setup\Products\ConfigurationTool”文件夹下，双击“setup. exe”，按照提示安装即可。

安装“WinCC Configuration Tool”后，打开 Microsoft Excel，其菜单栏中新增了“WinCC”菜单，如图 3-43 所示。单击“WinCC”菜单中的“Create project folder”，将打开“New project folder”对话框，如图 3-44 所示。在该对话框中，可以建立一个新的 WinCC 项目的连接（“Establish connection to new project”），也可以建立已存在的 WinCC 项目的连接（“Establish connection to existing project”）。如果 WinCC 处于打开状态且存在打开的项目，在对话框中选择“Establish connection to existing project”选项并单击“continue”按钮，则在接下来的对话框中自动显示 WinCC 已经打开的项目的路径。单击“Complete（完成）”按钮，即可在 Microsoft Excel 中打开此 WinCC 项目中的列表。选择“Tags”选项，WinCC 项目变量便导出到 Microsoft Excel 列表中。




图 3-43　Microsoft Excel 中新增的 WinCC 菜单

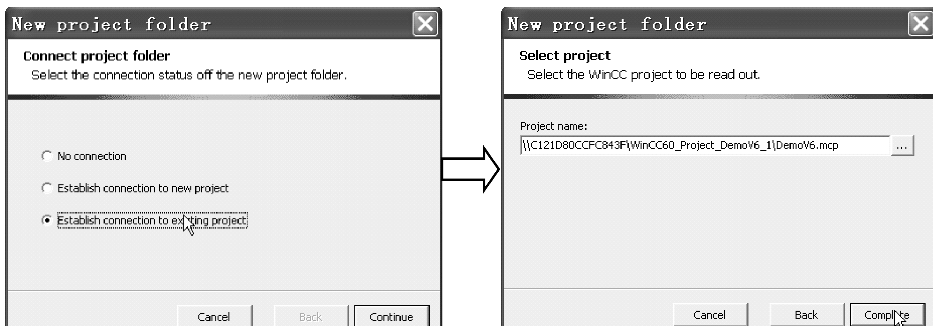



图 3-44　New Project folder 对话框

## 3.3.7　诊断 WinCC 变量状态的方法

在 WinCC 处于运行状态时，可以通过在变量管理器中查询变量的质量代码和变量的最后修改时间等信息来诊断变量的状态。

在 WinCC 项目激活状态下，在变量管理器中用鼠标指向要诊断的变量，就会出现一个提示框，如图 3-45 所示，提示框中显示了该变量的过程值、质量代码和最后修改时间等信息。通过质量代码可以诊断该变量的状态。如果质量代码为 80（质量代码是一个十六进制的数值），则表示变量的状态正常；如果质量代码为其他值，则可以在 WinCC 的帮助系统中查找变量的质量代码的相关信息，确定变量状态不正常的原因，如图 3-46 所示。




图 3-45　诊断变量的状态




图 3-46　变量的质量代码

# 第4章

# 过程画面的组态及用户权限管理

过程可视化是 WinCC 最基本、最重要的功能，而实现过程可视化的关键是过程画面的组态。本章将详细介绍用 WinCC 图形编辑器组态过程画面的方法和技巧。用户权限管理用于组态过程画面和画面对象的访问权限，本章将详细介绍用户权限管理的基本概念以及组态方法。

## 4.1 WinCC 图形编辑器

WinCC 图形编辑器是用于创建过程画面并使其动态化的编辑器。只有在 WinCC 项目管理器中打开的项目，才能启动图形编辑器进行过程画面的组态。WinCC 项目管理器可以用来显示当前项目中可用画面的总览。WinCC 图形编辑器所编辑的画面文件的扩展名为 .PDL，WinCC 项目中所创建的画面保存在项目文件夹的“GraCS”文件夹中，项目之间可以通过复制此文件夹共享画面文件。

注意：高版本 WinCC 所创建的过程画面在低版本的 WinCC 中不可用。

在 WinCC 项目管理器的浏览窗口用鼠标右键单击“图形编辑器”，在弹出的菜单中选择“新建画面”，将创建一个新的画面，画面文件名称在数据窗口中显示。右键单击画面名称，将弹出一个下拉菜单，在下拉菜单中可以选择“重命名画面”修改画面的名称，选择“定义画面为启动画面”可以设置图形运行系统的起始画面，激活 WinCC 运行系统时将首先进入起始画面。

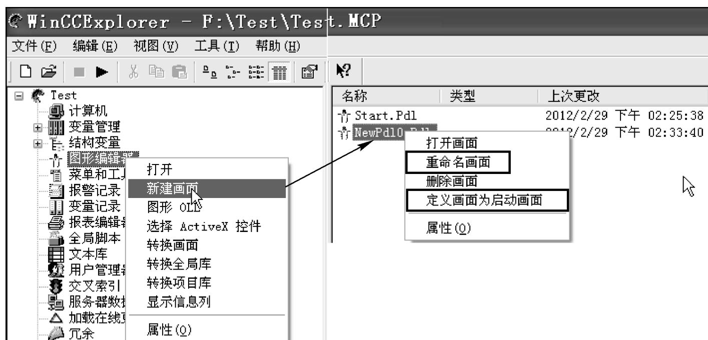



图 4-1　新建画面

### 4.1.1　图形编辑器的组成

双击图形画面名称，或在浏览窗口双击“图形编辑器”等可以进入图形编辑器的组态界面。图形编辑器组态界面中集成了图形画面编辑的常用工具和选项板，主要包括调色板、缩放工具、图层、对齐板、对象选项板和动态向导等，如图 4-2 所示。在图形编辑器的菜单栏中选择“视图”→“工具栏”，在弹出的“工具栏”对话框中可以设置打开或隐藏的各种工具和选项板。




图 4-2　图形编辑器

1. 绘图区

绘图区位于图形编辑器的中央。在绘图区中，水平方向为 X 轴，垂直方向为 Y 轴，画面左上角为坐标原点（X = 0，Y = 0），坐标以像素为单位。

在绘图区添加的画面对象的坐标原点是该对象矩形轮廓的左上角。

2. 对象选项板

对象选项板中包含了在组态过程画面过程中频繁使用的各种类型的画面对象，包括“标准”和“控件”两个选项卡，如图 4-3 所示。

（1）“标准”选项卡

“标准”选项卡中包含三类画面对象：标准对象、智能对象和 Windows 对象。

- 标准对象：包含用于生成复杂图形的各种基本图形以及可以作为生成对象的文字标题的静态文本。
- 智能对象：提供了创建复杂过程画面的画面对象，这些对象预定义了一些属性，用户只需设置相应的参数或进行相应的组态就可以使其动态化。智能对象是组态过程画面过程中最常用的一类画面对象之一。各种智能对象详细的功能描述见表 4-1。



- Windows 对象：提供了类似于按钮、复选框、选项组和滚动条等 Windows 应用程序的元素。可以采用多种方法对 Windows 对象进行编辑并使其动态化，用户可以通过 Windows 对象实现过程事件的操作和过程的控制。各 Windows 对象详细的功能描述见表 4-2。

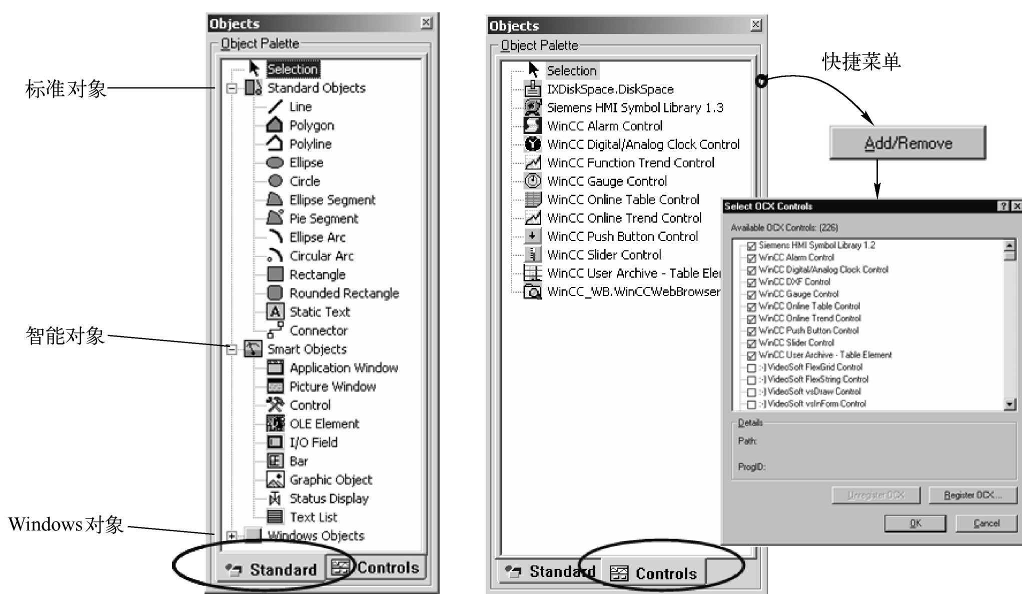



图 4-3　图形编辑器的对象选项板

**表 4-1　智能对象功能描述**

| 图　　标 | 对　　象 | 描　　述 |
|---|---|---|
| | 应用程序窗口 | 应用程序窗口是由其他应用程序（如报警记录、变量记录、打印作业和全局脚本）管理的窗口对象。在运行系统中，应用程序窗口将打开这些应用程序，并传送相关信息以及允许相关的操作。应用程序窗口在运行时的大小和属性在图形编辑器中定义 |
| | 画面窗口 | 画面窗口可以在当前画面中显示其他的用图形编辑器创建的画面。画面窗口中显示的画面可以用动态进行连续地更新。画面窗口在运行时的大小和属性在图形编辑器中定义 |
| | 控件 | 控件是一些提供预定义功能的对象，例如报警窗口、测量窗口、选择对话框或按钮等。控件对象允许插入在操作系统中注册过的 Active 控件、WinCC 控件和来自其他制造商的控件。这些控件根据需要改变并动态地集成到过程中。控件在运行时接受的大小和属性在图形编辑器中定义 |
| | OLE | OLE 允许将在其他程序中创建的文件插入到 WinCC 画面中，因此可集成所有在操作系统中注册的 OLE 对象。OLE 对象在运行系统中接受的大小和属性在图形编辑器中定义。在运行系统中不能对 OLE 对象进行改变 |
| | I/O 域 | I/O 域可以定义为输入域、输出域或输入/输出域的组合。可用下列数据格式：二进制、十进制、字符串和十六进制。也可以指定诸如“隐藏输入”或“在全部输入时接受”等限制值。I/O 域在运行时的大小和属性在图形编辑器中定义 |

（续）

| 图　　标 | 对　　象 | 描　　述 |
|---|---|---|
| | 棒图 | 棒图提供图形化显示数值的功能。数值也可以在组合视图中以带有可自由定义的数字刻度的图形显示。棒图在运行时的大小和属性在图形编辑器中定义 |
| | 图形 | 图形允许在画面中插入其他程序中创建的图形。可插入具有下列格式的画面和图形：EMF、WMF、DIB 和 BMP。图形对象在运行时的大小和属性在图形编辑器中定义 |
| | 状态显示 | 状态显示可以显示一个对象的几乎任意数目的不同状态。状态的变化通过变量值的变化实现，每一个变量值对应一个分配的画面来显示对象的一个状态。状态显示在运行系统中的大小和属性在图形编辑器中定义 |
| | 文本列表 | 文本列表提供分配指定数值给文本的功能。文本列表可定义为输入列表、输出列表或组合的输入/输出列表。可用下列数据格式：十进制、二进制或位。文本列表在运行时的大小和属性在图形编辑器中定义 |
| | 3D 棒图 | 3D 棒图允许以三维图形方式显示数值。3D 棒图在运行系统中的大小和属性在图形编辑器中定义。只有“基本过程控制”可选软件包随 WinCC 一起安装才可用 3D 棒图 |
| | 组显示 | 组显示提供按层级显示特定消息类型的当前状态的功能。四个按钮允许显示和操作消息。例如，使用组显示，能够实现快速地切换以显示错误源。组显示在运行系统中的大小和属性在图形编辑器中定义。只有“基本过程控制”可选软件包随 WinCC 一起安装才可用组显示 |

表 4-2　Windows 对象功能描述

| 图　　标 | 对　　象 | 描　　述 |
|---|---|---|
| | 按钮 | 按钮允许过程操作。例如，可用于确认消息。按钮可有两种状态：“开”和“关”。通过提供动态给相应的属性，将按钮集成到过程中 |
| | 复选框 | 复选框允许操作员选择多个选项并且选中相应的复选框。也可以在默认状态下选中复选框，这样操作员只要根据需要改变预定义的数值即可。通过提供动态给相应的属性，复选框可集成到过程中 |
| | 选项框 | 选项框允许操作员选择所显示选项中的一个，并激活相应的选项域。也可以在默认状态下激活选项域中的一个，这样操作员只要根据需要改变预定义的数值即可。通过提供动态给相应的属性，选项框可集成到过程中 |
| | 圆形按钮 | 圆形按钮允许过程操作。例如，用于在运行时确认消息或浏览。圆形按钮可有三种状态：“开”、“关”和“取消激活”。也可以定义圆形按钮是否应该在默认状态下显示为已按下，是否该在按下时单击。通过提供动态给相应的属性，圆形按钮可集成到过程中 |
| | 滚动条 | 滚动条是个滚动控制器，可用来控制过程。例如，用于变量无限地改变到某一数值。通过提供动态给相应的属性，滚动条可集成到过程中 |

（2）“控件”选项卡

“控件”选项卡中包含由 WinCC 提供的最重要的 ActiveX 控件，也可以使用在 Windows 系统中注册的，通过注册还可以使用第三方供应商的 ActiveX 控件。表 4-3 是 WinCC 提供的常用的 ActiveX 控件。



表 4-3　常用的 WinCC ActiveX 控件

| 名　　称 | 功　　能 |
|---|---|
| WinCC Digital/Analog Clock Control | WinCC 数字/模拟时钟控件可在运行系统中显示操作系统的当前系统时间。时间可显示为模拟或数字式时间 |
| WinCC Gauge Control | WinCC 量表控件用于显示以模拟量表形式表示的监控测量值。警告和危险区域以及指针运动的极限值均用颜色进行了标记 |
| WinCCOnline Table Control | WinCC 在线表格控件用于以表格形式显示过程值的当前值或归档值 |
| WinCC Online Trend Control | WinCC 在线趋势控件用于以趋势图的形式显示过程值的当前值或归档值 |
| WinCC Push Button Control | WinCC 按钮控件可用于组态与命令执行相关联的命令按钮 |
| WinCC Slider Control | WinCC 滚动条控件可用于显示以滚动条控件形式表示的监控测量值。当前值可显示在滚动条下方，受控测量区则可显示为刻度标签 |
| WinCC User Archive - Table Element | WinCC 用户归档表格元素提供了访问用户归档和用户归档视图的选项 |
| WinCC Alarm Control | WinCC 报警控件用做运行系统中的一个消息窗口来显示消息事件 |
| WinCC Function Trend Control | WinCC 函数趋势控件用于显示随其他变量改变的变量的数值，并将该趋势与设定值趋势进行比较 |

在对象选项板中可以添加或删除 ActiveX 控件，如图 4-3 所示，选中“控件”选项卡，用鼠标右键单击“选择”，然后选择弹出的菜单“添加/删除”，打开“选择 OCX 控件”对话框，在可用的控件列表中，红色复选标记标识的控件表示该控件在对象选项板的“控件”选项卡中可用，因此可以激活/取消激活红色复选标记来添加/删除 ActiveX 控件。

### 3. 样式选项板

样式选项板允许快速更改线型、线条粗细、线端和填充图案，包括的样式组有：

- “线型”样式组：包含不同的线条显示选项，如虚线、点划线等。
- “线粗细”样式组：可以设置线的宽度，线粗细按像素指定。
- “线端样式”样式组：允许显示线尾端的形状，如箭头或圆形。
- “填充图案”样式组：可以为封闭对象选择透明背景或实心背景以及各种填充图案。

### 4. 动态向导

动态向导如图 4-4 所示，动态向导提供大量的预定义的 C 动作，可以简化频繁重复出现的过程的组态。动态向导按类分为多个选项卡，各选项卡中包含不同的 C 动作。

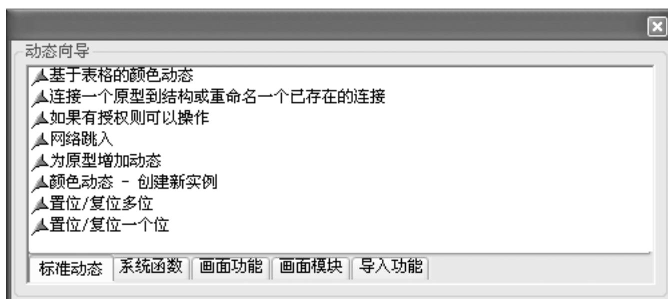



图 4-4　图形编辑器的动态向导

### 5. 对齐选项板

对齐选项板可用于同时处理多个对象。这些功能也可以从“排列”→“对齐”菜单

中调用。主要功能包括：

- 对齐：所选择的对象向左、向右、向上或向下对齐。
- 居中：所选择的对象水平或垂直居中。
- 分散：所选择的对象水平或垂直分散。最外面对象的位置保持不变。各个对象相互平均地间隔。
- 调整宽度和高度：所选对象在宽度或高度上调整为一致。

6. 图层（Layer）选项板

为了简化在复杂的过程画面中处理单个对象，图形编辑器允许使用图层。例如，过程画面的内容最多可以横向分配为 32 个图层。这些图层可以单独地显示或隐藏，默认设置为所有图层均可见，激活的图层为图层 0。

7. 调色板

根据所选择的对象，调色板允许快速更改线或填充颜色。它提供 16 种标准颜色，单击调色板底部的按钮，可以打开“颜色”对话框，对话框中提供了 48 种基本颜色，并可以创建用户定义的颜色。

## 4.1.2　图形编辑器的基本操作

1. 公共属性设置

在组态过程画面之前，为了提高过程画面的质量以及运行系统的效率，需要对图形编辑器的一些公共属性进行设置。在图形编辑器的菜单栏中选择“工具”→“设置”，打开图形编辑器的“设置”对话框，如图 4-5 所示。

图 4-5 左图为“设置”对话框的“网格”选项卡，可以设置图形编辑器绘图区的网格。系统默认的网格像素为 10，为了在设计过程画面时更加精细，可以将网格像素的值减小。

图 4-5 右图为“设置”对话框的“缺省对象设置”选项卡，可以设置过程画面中画面对象的缺省触发器。触发器用于在运行系统中执行画面或画面对象的动态，因此，将触发器与对象的动态相链接以构成对动态进行调用的触发事件，没有触发器的动态不会执行。除非在组态过程中为某个特定的画面对象单独分配一个触发器，否则所有画面对象在运行系统中以缺省触发器执行动态。在“设置”对话框中修改缺省触发器，只对修改后组态的画面对象有效，修改前组态的画面对象仍保持原来的缺省触发器。

2. 导出功能

在图形编辑器中，导出功能位于“文件”菜单下，可将画面或选择的画面对象导出到其他的文件中。画面或画面对象可以从图形编辑器以 EMF（增强型图元文件）和 WMF（Windows 图元文件）文件格式导出。然而，在这种情况下，动态设置和一些对象特定属性将丢失，因为图形格式不支持这些属性。

还可以以程序自身的 PDL 格式导出图形。然而，以 PDL 格式只能导出整个画面，而不能是单个画面对象。另一方面，画面导出为 PDL 文件时，动态设置得以保留，导出的画面可以插入画面窗口中，也可以作为画面文件打开。

3. 导入功能

在图形编辑器中，导入功能位于“文件”菜单下，可将其他程序创建的图形作为画面对象、OLE 对象或可编辑图形插入图形编辑器中。可插入的图形或图像格式有：EMF、



WMF、DIB、BMP、GIF、JPEG、ICO 和 PNG 等。而可编辑图形必须是以 EMF 或 WMF 格式保存的向量图形。




图 4-5　图形编辑器的“设置”对话框

**4. 激活 WinCC 运行系统**

在图形编辑器中也可以激活 WinCC 运行系统，单击菜单“文件”→“激活运行系统”，或单击工具栏的▶按钮均可以实现。在图形编辑器中激活运行系统，首先激活的画面是图形编辑器当前打开的过程画面，而项目管理器中激活运行系统始终以项目的起始画面为起点。

可以在 WinCC 项目处于激活状态时对过程画面进行编辑和修改，编辑和修改后对画面进行保存，然后再次在图形编辑器中激活运行系统，便可以显示编辑和修改后的画面。

**5. 画面对象的基本静态操作**

（1）将画面对象插入画面

这里的画面对象指的是图形编辑器的对象选项板中包含的标准对象、智能对象、Windows 对象和控件，插入到画面的步骤如下：

1）打开需要插入对象的画面。

2）在对象选项板上单击需要插入的对象。

3）将光标定位在画面上，鼠标指针变成带有一个要添加对象符号的十字准线。

4）在插入的位置（画面对象的左上角）单击并拖动鼠标左键，插入的对象包含在一个矩形框中，拖动矩形到所需大小，释放鼠标，对象被插入到画面。

也可以在对象选项板上双击需要添加的画面对象，画面对象将以默认的大小插入到画面中。

（2）更改画面对象的静态属性

对象的静态属性是改变对象的静态数值，定义如对象的形状、外观、位置或可操作性等。

不同类型的画面对象有不同的属性选项，画面对象被插入到画面后，可以对画面对象的静态属性进行更改。用鼠标右键单击需要更改属性的画面对象，在弹出的菜单中选择“属性”，将打开“对象属性”窗口，如图 4-6 所示。“对象属性”窗口左边列表框列出了画面对象的属性选项，右边列表框的“静态”列可以对所选属性选项中各属性的静态值进行更改。




图 4-6　更改画面对象的静态属性

对象的属性分为静态属性和动态属性。静态属性是指组态时设置的对象属性在系统运行过程中保持不变。动态属性是指组态时设置的对象属性要组态动态，在运行状态下当组态的动态发生时对象的属性将发生变化。对象的某个属性组态动态后，其属性名称将以黑体字显示，对象的动态组态方法将在 4.3.3 小节中介绍。

（3）更改画面对象的默认静态属性

将画面对象插入到画面中时，该画面对象的多数静态属性值是系统的默认值，如图 4-6 中插入的“输入输出域 1”的“字体”→“X 对齐”属性值为“左”。如果组态过程中某个类型画面对象需要插入多个，且某些属性的值不采用默认值，除了可以通过复制对象（被复制的对象的属性值已更改）的方法外，还可以采用更改该类型画面对象的默认对象属性的方法。

用鼠标右键单击对象选项板中需要修改默认属性的画面对象，在弹出的菜单中选择“属性”，打开“对象属性”窗口，更改期望修改的对象属性。后面插入到画面中的该类型画面对象的相关属性都采用修改后的值。

（4）选择多个画面对象

选择多个画面对象可以通过鼠标的框选或借助键盘的〈Shift〉键和鼠标实现。

- 鼠标框选：在画面的合适位置单击，拖动鼠标，形成的矩形框将发生变化，到矩形框包含了所有需要选择的画面对象时释放鼠标左键，这些画面对象将被选中。
- 〈Shift〉键 + 鼠标：按住键盘的〈Shift〉键，同时鼠标左键一个接一个地单击所要选择的画面对象。

（5）组对象

当需要将多个对象当作一个整体使用时，可以使用组对象。在所有需要编组的画面对象被选中的状态下，用鼠标右键单击任意一个被选中的画面对象，在弹出的菜单中选



择“组对象”→“编组”，此时所有选中的对象就成为一个对象了，如图 4-7 所示。

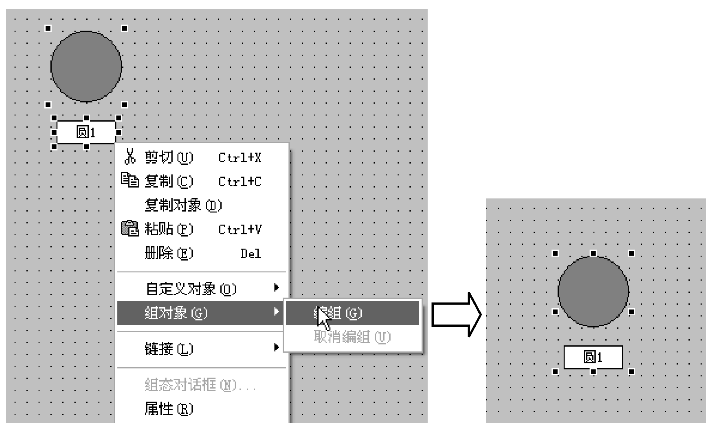



图 4-7　组对象

## 4.1.3　图形编辑器的库

图形编辑器的库是一种用于保存和管理创建过程画面中所使用的图形对象的通用工具，分为全局库和项目库两个区域。

“全局库”提供了多种预先完成的图形对象，这些对象可作为库对象插入到过程画面中，并根据需要进行组态。“全局库”以树结构形式按主题进行排序。使用用户自定义的对象可扩展“全局库”，以便使其适用于其他项目。但是，这些对象不一定要动态链接，以避免在将其嵌入到其他项目中时出现错误。

“项目库”允许建造一个项目特定的库。通过创建文件夹和子文件夹可按主题对对象进行排序。“项目库”可将用户自定义对象作为副本进行存储，使其可以多次调用。因为“项目库”只适用于当前项目，因此，动态对象应只放在该库中，插入到该库中的自定义对象的名称可自由选取。

单击图形编辑器工具栏中的“显示库”按钮，或在菜单栏中选择“视图”→“库”，打开图形编辑器的库，如图 4-8 所示。选中某个库，单击库工具栏中的“超大图标”按钮和“预览”按钮，在库的右侧窗口中将显示该库中的所有对象。

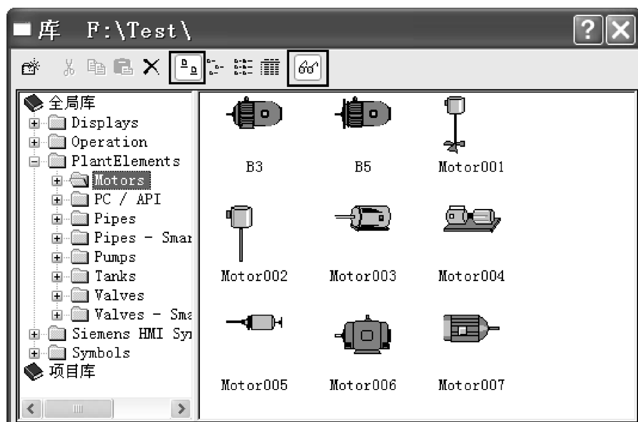



图 4-8　图形编辑器的库

## 4.2　动态变化的触发器和更新周期

创建过程画面最重要的工作是组态过程画面和画面对象的动态属性，从而使得过程画面和画面中的对象可以反映过程的变化，这称为画面和画面对象的动态变化。组态动态变化时必须涉及触发器和更新周期的概念，指定触发器和更新周期是组态系统的重要设置，它影响画面、画面对象的更新、后台脚本的处理等。当确定触发器和更新周期时，要从整体上考虑，不合适的触发器和更新周期对 HMI 系统的性能有负面的影响。

图 4-9 列出了 WinCC 的触发器类型，相关概念如下：

- 周期性触发器：是 WinCC 中处理周期性动作的方法，周期性触发器的动作将在固定时间间隔内重复执行。周期性触发器的第一个时间间隔的起始点要与运行系统的起始点一致，间隔时间由周期确定。可选择 250 ms 和 1 h 之间的周期，也可使用自定义的用户周期。对于过程画面和画面对象，还可以选择基于窗口周期的周期性触发器和基于画面周期的周期性触发器。
- 非周期性触发器：非周期性触发器的动作只执行一次，起始点由日期/时间确定。
- 变量触发器：变量触发器由一个或多个指定的变量组成。如果这些变量其中一个的数值的变化在启动查询时被检测到，则与这样的触发器相连接的动作将执行。“循环监视变量值”是按一定的时间间隔查询变量的值，第一个时间间隔的起始点与运行系统的起始点一致，间隔时间由周期确定；“根据变化”无论变量的值何时发生变化，该触发器关联的动作都执行。对于过程变量，“根据变化”模式相当于一个有 1 s 周期的循环读作业。
- 事件驱动的触发器：只要事件一发生，与该事件相连接的动作就将执行。例如，事件可以是鼠标控制、键盘控制、或焦点的变化等。

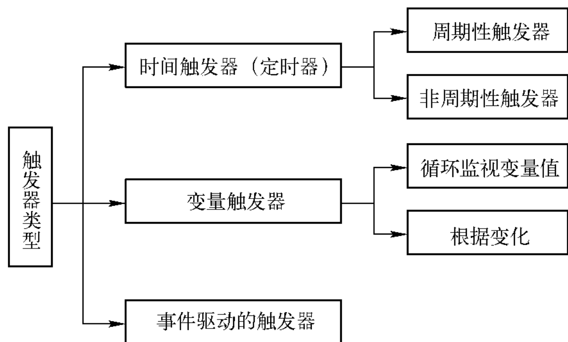



图 4-9　触发器类型图

- 画面周期：将周期性的触发器用做触发器。周期时间由画面的属性“更新周期”定义。该周期提供了一个选项，可集中定义在画面中使用的所有动作的周期。
- 窗口周期：将周期性的触发器用做触发器。周期时间由“画面窗口”对象的“更新周期”对象属性定义。该周期提供了一个选项，可集中定义在“画面窗口”对象中使用的所有动作的周期。



# 4.3　组态过程画面

## 4.3.1　设计过程画面的结构

一个 WinCC 项目一般包含多幅过程画面，且各个画面之间应该能按要求互相切换。根据项目的总体要求和规划，首先需要对过程画面的总体结构进行设计，确定需要创建的过程画面以及每个画面的主要功能等。其次需要分析各个画面之间的关系，并根据操作的需要安排画面之间的切换顺序。各画面之间的相互关系应该层次分明、操作方便。

例如，经过分析，为一个物料灌装自动生产线的 HMI 项目设计了 5 幅过程画面，分别是起始画面、运行画面、参数设置画面、趋势视图画面和报警画面。各画面的功能定义如下：

- 起始画面：激活 WinCC 项目时首先显示的画面，该画面显示了项目的一些信息，经过授权的用户可以从起始画面切换到所有其他的画面。
- 运行画面：运行画面可以显示灌装自动生产线的状态，当生产线当前为远程控制方式（即由 HMI 系统进行控制）时，也可以通过画面中的画面对象对生产线进行控制。
- 参数设置画面：用来设置生产线中根据工艺的不同需要修改变化的参数，如生产线运行时间、限制值等参数。
- 趋势视图画面：用于显示生产线的一些过程值的变化曲线，如物料温度、流量和液罐的液位等过程值的变化曲线，用户可以根据变化曲线进行分析。
- 报警画面：实时显示反映生产线的运行状态的故障消息文本和报警记录。

## 4.3.2　设计过程画面的布局

图形编辑器绘图区的任何区域都可以组态各种对象和控件。为了方便监视和控制过程，通常可将过程画面的布局分为 3 个区域：概览区、工作区和键集合区，用户可以根据需要进行布局设计。如图 4-10 为典型的过程画面布局，左图的过程画面包含概览区（1）、工作区（2）和键集合区（3），而右图的过程画面仅包含工作区（2）和键集合区（3）两个区域。

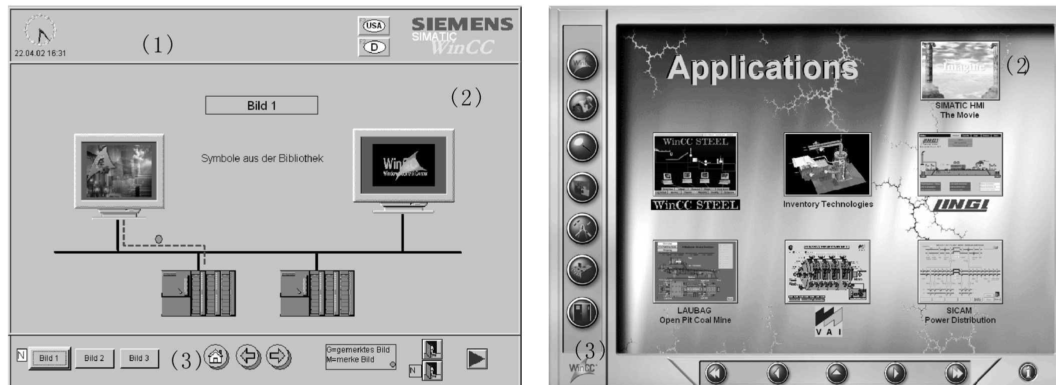



图 4-10　典型的过程画面布局

过程画面各区域的功能如下：

- 概览区：通常包括在所有画面都显示的信息，例如项目标志、项目运行时间和日期、报警消息和系统信息等。
- 工作区：用于显示过程事件。
- 键集合区：用于显示可以操作的按钮，例如工作区域的切换按钮、调用信息按钮等。按钮可以独立于所选择的工作区域使用。支持鼠标和/或功能键的操作。

常用的画面布局示意图如图 4-11 所示。

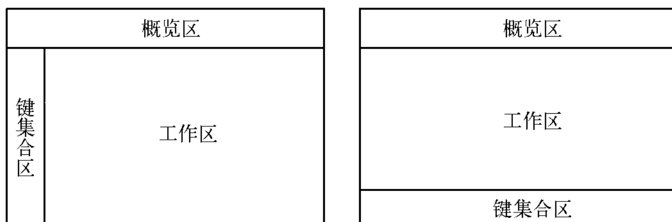



图 4-11　常用的画面布局示意图

## 4.3.3　创建过程画面

以 4.3.1 小节提及的物料灌装自动生产线的 HMI 项目为例，组态 5 个过程画面，组态的详细步骤如下：

1）组态 5 个过程画面。参考图 4-1 介绍的方法新建 5 个过程画面，重命名 5 个画面分别为“起始画面”、“运行画面”、“参数设置画面”、“趋势视图画面” 和 “报警画面”，并将名称为“起始画面” 的过程画面定义为启动画面，如图 4-12 所示。

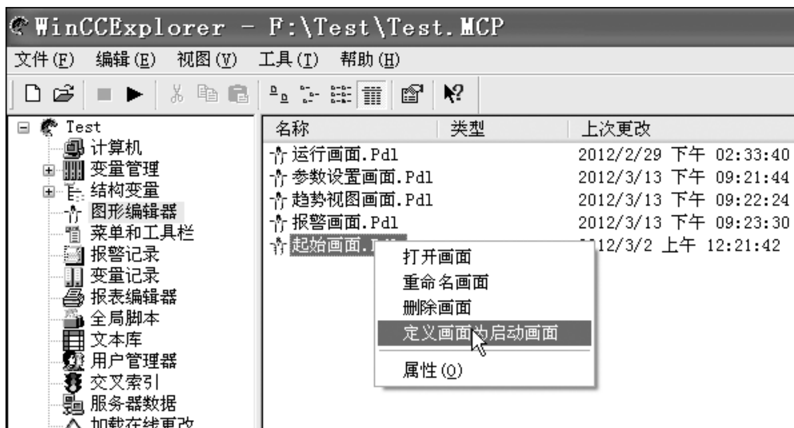



图 4-12　新建 5 个过程画面

2）双击“初始画面”，进入图形编辑器，同时打开“初始画面”。设置画面的几何属性，宽度 1280、高度 800（该属性根据当前计算机的屏幕分辨率设置）。

3）在初始画面的概览区，添加一个“WinCC 时钟控件”，用于在运行时显示系统的当前时间。在对象选项板的“控件”选项卡中，将“WinCC Digital/Analog Clock Control”



添加到绘图区。用鼠标右键单击添加的时钟，在弹出的菜单中选择“属性”，打开“对象属性”对话框，如图 4-13 所示。将“模拟”属性设为“否”，以显示数字时钟，更改“字体”属性至合适的字体、字形和大小。




图 4-13　设置时钟控件的属性

4）从对象选项板的“标准”选项卡中打开“标准对象”选项卡，将“静态文本”添加到起始画面的工作区，并设置“静态文本”的属性，如图 4-14 所示。

5）从对象选项板的“标准”选项卡中选择“智能对象”→“图形对象”，添加到起始画面的工作区，在弹出的“图形对象组态”窗口中单击“查找”命令，选择已经保存在计算机上的图形或图片到“图形对象”中显示，如图 4-15 所示。

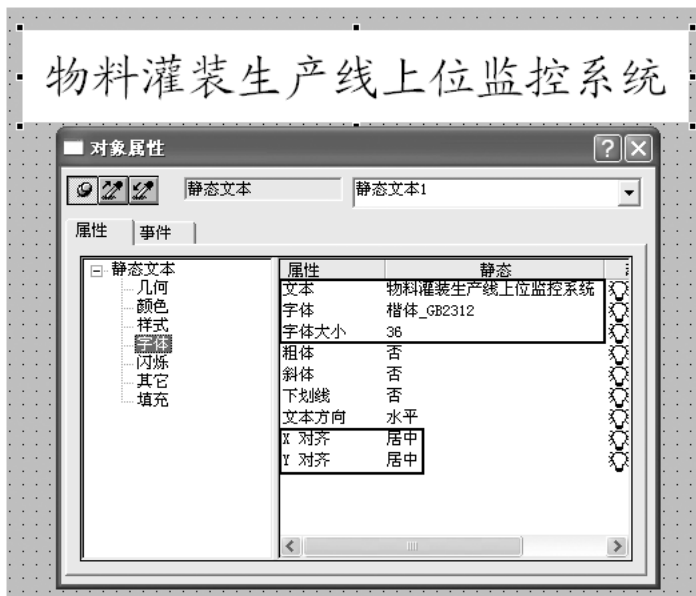



图 4-14　设置“静态文本”的属性







图 4-15　添加图片到“图形对象”

其他过程画面可以参考上述的方法添加所需的画面对象到画面中，并编辑相关的属性，也可以在组态动态的过程中随时进行画面对象的添加和编辑。

### 4.3.4　过程画面的动态化

组态过程画面以及画面中的对象的动态是组态过程画面最重要的工作，这里的动态指的是在运行状态时过程画面或画面中的对象通过其物理属性（例如颜色、几何尺寸、位置等属性）的变化反映过程的状态，或是通过操作画面中的对象（例如鼠标单击按钮）来控制过程。

**1. 动态的分类**

（1）属性动态

指画面或画面对象的物理属性如颜色、对象的尺寸、位置及字体大小等，在运行时发生动态变化。“对象属性”窗口中的“属性”选项卡中允许组态属性动态化，如图 4-16 所示。窗口的左侧区域以目录树的形式显示所选对象及其属性组；右侧区域包含可用于所选属性组的所有属性，显示有“属性”、“静态”、“动态”、“更新周期”和“间接”5 列。

- “属性”列：显示属性的名称，所选属性组中可用于所选对象的所有属性均将显示。不能改变属性的名称。属性的静态值可通过双击属性名称改变。
- “静态”列：显示所选对象属性的当前值。根据属性类型，值将显示为数字、文本或图形。如果“动态”列没有组态，则在运行时对象的属性为设定的静态值。
- “动态”列：设置对象属性的动态化。如果组态了对象的动态属性，则在运行时对象属性可以动态变化。“动态”列显示为白色灯泡时表示没有组态动态。用鼠标右键单击白色灯泡，在弹出的菜单中可以选择动态化的方法组态该属性的动态，如图 4-12 所示。
- “更新周期”列：如果组态了属性的动态，“更新周期”列将显示设置的动态更新周期，双击数值，可以改变属性的更新周期。
- “间接”列：属性可以通过直接或间接的方式实现动态化。






图 4-16 “对象属性”窗口的“属性”选项卡

（2）事件动作

在运行状态时，画面或画面对象可以响应由系统或操作员产生的一些事件，如果针对一个对象的某个事件组态一个动作，则当该事件产生时，该对象将执行组态的动作。“对象属性”窗口中的“事件”选项卡中允许对对象的事件进行组态，如图 4-17 所示，窗口左侧区域以目录树的形式显示所选对象可以响应的事件组，右侧区域包含了所选事件组的所有事件。“动作”列显示为白色闪电时表示没有组态动作，用鼠标右键单击白色闪电，在弹出的菜单中选择动态化方法可以组态该事件的动作。




图 4-17 “对象属性”窗口中的“事件”选项卡

## 2. 组态动态化的方法

WinCC 提供了多种组态过程画面动态化的方法，包括组态对话框、动态对话框、动

态向导、变量连接、直接连接、C 脚本和 VB 脚本等，见表 4-4。将在 4.4 节中详细介绍这些方法的概念和应用。

表 4-4　组态过程画面动态化的方法

| 方　法 | 说　明 | 类　型 | 显示标志 |
|---|---|---|---|
| 组态对话框（快速组态） | 支持智能对象和 Windows 对象的组态 | I/O | 属性动态——绿色灯泡（变量连接）<br>事件动作——蓝色闪电（直接连接） |
| 动态向导（组态助手） | 生成 C 脚本，可组态复杂功能 | I/O | 绿色闪电<br>黄色闪电（未通过编译） |
| 变量连接（1 对 1 的连接） | 属性动态 | O | 绿色灯泡 |
| 动态对话框（复杂的连接） | 属性动态 | O | 红色闪电 |
| 直接连接 | 事件动作 | I | 蓝色闪电 |
| C 动作 | 属性动态 | O | 绿色闪电<br>黄色闪电（未通过编译） |
| | 事件动作 | I | |
| VBS 动作 | 属性动态 | O | 浅蓝色闪电 |
| | 事件动作 | I/O | |

# 4.4　组态动态化举例

## 4.4.1　组态对话框

“组态对话框”（Configuration Dialog）是 WinCC 的图形编辑器提供的快速组态工具，当具有组态对话框功能的画面对象插入到绘图区时，“组态对话框”窗口将会自动弹出，用户可以快捷地为该对象的一些特定属性组态动态，设置属性参数等。

**1. 支持组态对话框功能的画面对象**

具有组态对话框功能的对象有：

- 智能对象中有控件、输入/输出（I/O）域、棒图、图形对象、状态显示、文本列表。
- Windows 对象有按钮和滚动条。

注意：在图形编辑器的菜单栏中，选择“工具”→“设置”→“选项”命令，激活“使用组态对话框”，这样才能在插入智能对象和 Windows（窗口）对象时自动弹出“组态对话框”，或在用鼠标右键单击对象时能够选择“组态对话框”。

**2. 组态实例**

（1）组态可以实现画面切换的按钮

为了实现在运行状态下各个画面之间可以切换显示，需要在各自画面的键集合区组态画面切换的按钮。

以 4.3.3 小节创建的 5 个过程画面为例，在起始画面中，选择“对象选项板”→“窗口对象”→“按钮”，添加到起始画面的键集合区，在弹出的“按钮组态”对话框中（该对话框即为按钮对象的组态对话框）输入文本“报警画面”，并设置字体、颜色等属



性。单击按钮，在打开的“画面”窗口中选择单击该按钮时要打开的画面，如图 4-18 中“报警画面”按钮对应切换到“报警画面 . Pdl”。用相同的方法添加“运行画面”、“趋势视图”等按钮，并组态相应的切换功能。




图 4-18　组态画面切换的按钮

打开其他的画面，根据画面之间的切换关系，可按上述方法组态相应的切换按钮，也可以直接复制起始画面中已经组态的切换按钮。例如，假定运行画面和报警画面之间需要实现相互切换，可将起始画面中组态的标有“运行画面”的按钮复制到报警画面，将标有“报警画面”的按钮复制到运行画面中。

保存所有画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试画面切换的结果。

（2）组态 I/O 域以数字形式显示过程值

I/O 域可定义为输入域、输出域或输入/输出域的组合，这里的输入指的是输入数据来更改 I/O 域所连接的变量的值，输出指的是显示 I/O 域所连接的变量的值，数据格式可以是二进制、十进制、字符串和十六进制。

在图形编辑器中，选择“对象选项板”→“智能对象”→“输入/输出域”，添加到画面的工作区，将自动弹出“I/O 域组态”对话框，如图 4-19 所示。在对话框中单击按钮，在弹出的“变量 - 项目”窗口中选择需要连接的变量，并根据需要设置更新、I/O 域类型、字体大小、字体和字体颜色等属性。

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

注意：在组态过程画面时，每完成一个动态化的组态，都可以激活项目对组态的结果进行测试。用户无需取消激活状态，只需保存所完成的组态，重新执行一次激活项目




图 4-19　I/O 域的组态对话框

的操作即可。另外，测试组态结果多数情况是需要更改过程变量或内部变量的值，用户可以通过STEP 7提供的PLC仿真器（S7-PLCSIM）或WinCC提供的变量仿真器（WinCC TAG Simulator）仿真变量的变化，PLC仿真器和WinCC变量仿真器的用法请参考本章4.7.1小节。

（3）组态棒图图形化显示过程值

棒图对象提供图形化显示数值的功能。

在图形编辑器中，选择“对象选项板”→“智能对象”→“棒图”，添加到画面的工作区，将自动弹出“棒图组态”对话框，如图4-20所示。在对话框中单击按钮，在弹出的“变量-项目”窗口中选择需要连接的变量，设置棒图的最大值和最小值（需要根据所连接的变量进行设置）。根据需要，还可以在“对象属性”窗口中设置棒图的更多属性。

保存画面，在WinCC图形编辑器或项目管理器中激活项目，测试组态的结果。




图4-20　“棒图组态”对话框

## 4.4.2　动态向导

动态向导提供大量WinCC系统预定义的C动作，以支持频繁重复出现的过程的组态，提高了组态的效率，同时降低了发生组态错误的风险。用户可以用自己创建的函数对动态向导中的C动作进行扩展。

**1. 动态向导窗口**

在图形编辑器的菜单栏中选择“视图”→“工具栏”，在弹出的“工具栏”对话框中可以设置动态向导窗口的显示或隐藏，动态向导窗口可以在图形编辑器的任意位置显示。

动态向导窗口中包含“标准动态”、“系统函数”、“画面功能”、“画面模块”和“导入功能”5个选项卡，每个选项卡中包含有多个可以组态的动态向导。

**2. 组态实例**

（1）组态“退出WinCC运行系统”的按钮

参考第3章3.1.5小节，用户可以在WinCC项目管理器中单击■图标退出WinCC运



行系统，但是 WinCC 项目投入现场运行以后，除了有授权的工程人员外，一般不允许进入 WinCC 资源管理器。因此，有必要组态在运行系统的画面中可以退出 WinCC 运行系统的命令按钮（该按钮只能由具有授权的用户操作）。

下面通过动态向导的方法在 4.3.3 小节创建的“起始画面”中组态“退出 WinCC 运行系统”的按钮。

1）打开“起始画面”，选择添加“对象选项板”→“窗口对象”→“按钮”到画面的键集合区，并在自动弹出的“按钮组态”对话框中将按钮显示的文本修改为“退出 WinCC 运行系统”。

2）按钮处于选中状态时，在动态向导窗口选择“系统功能”选项卡，双击名称为“退出 WinCC 运行系统”的动态向导，如图 4-21 所示。

3）根据对话框中的向导提示完成需要的组态步骤。




图 4-21　组态“退出 WinCC 运行系统”的动态向导

打开该按钮的“对象属性”窗口并选择“事件”选项卡，可以看到组态动态向导后生成的 C 动作，如图 4-22 所示。

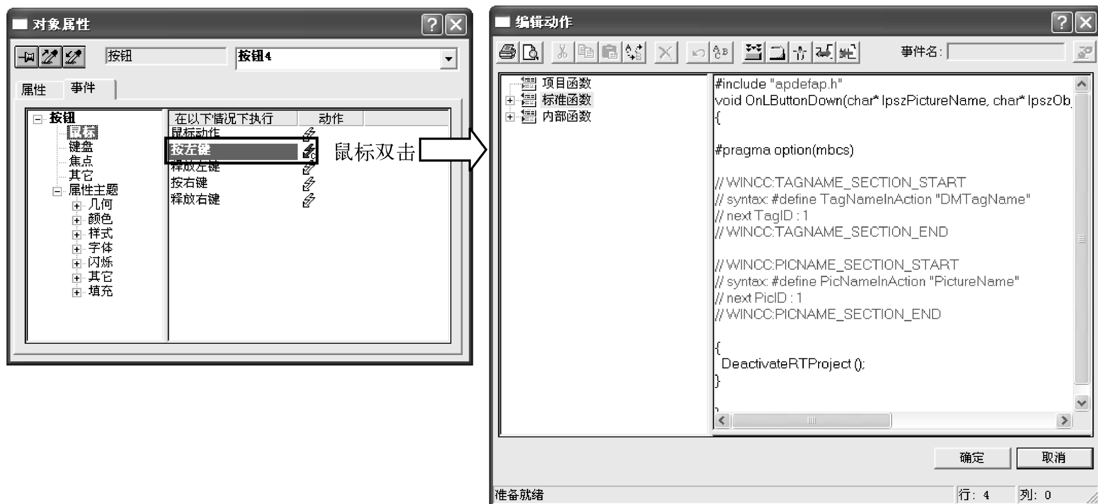



图 4-22　组态动态向导后生成的 C 动作

（2）组态“操作确认”对话框

为了防止误操作，在“起始画面”中组态“退出 WinCC 运行系统”按钮时，一般需

要组态一个“操作确认”对话框。单击“退出 WinCC 运行系统”按钮时，先弹出“操作确认”对话框，确认后按钮的动作才有效。“操作确认”对话框同样可以通过动态向导的方法组态。

1）“退出 WinCC 运行系统”按钮处于选中状态，在动态向导窗口的“图画功能”选项卡中，双击“显示错误框”命令，如图 4-23 所示。

2）按向导提示设置相应的参数，在“设置选项”对话框中，将“确认”对话框的标题设为“操作提示”，对话框消息文本设为“确认要退出运行系统吗?”，并激活带有“取消”按钮的框，如图 4-24 所示。

3）打开“退出 WinCC 运行系统”按钮的“对象属性”窗口并选择“事件”选项卡，按图 4-22 的方法双击生成的 C 动作，进入 C 动作编辑窗口，修改生成的 C 动作程序，如图 4-25 所示，矩形框内的程序如下：

```
if(MessageBox(NULL,"确认要退出运行系统?","操作提示",MB_OKCANCEL) == IDOK)
        DeactivateRTProject ( );
  else
        {
        }
```

4）编译修改后的程序，如果编译提示有错误，修改错误直至编译成功退出编辑窗口。




图 4-23　组态“确认”对话框的动态向导




图 4-24　设置“确认”对话框的动态向导参数



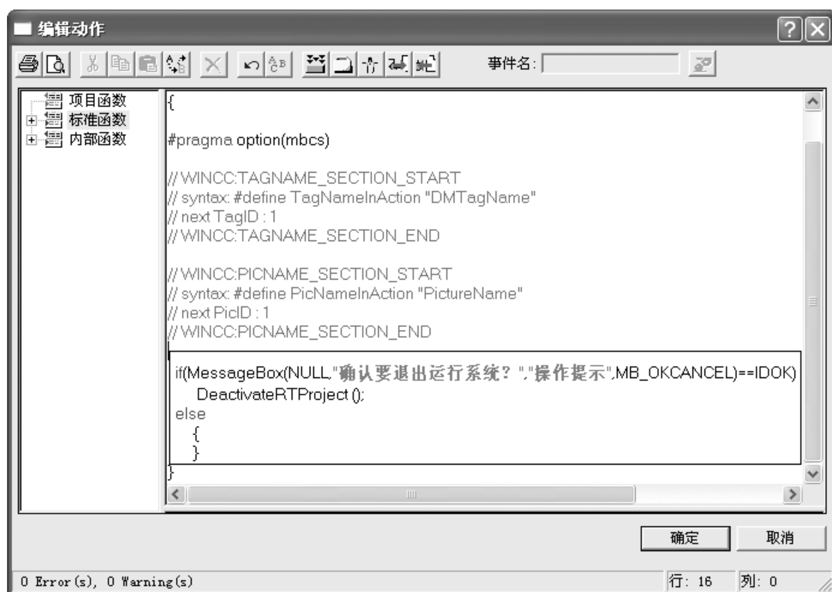



图 4-25　修改 C 动作程序

保存起始画面，激活项目后，单击起始画面中的“退出 WinCC 运行系统”按钮，弹出“确认”对话框，如图 4-26 所示，单击“确认”按钮，将退出 WinCC 运行系统，单击“取消”按钮，操作无效。




图 4-26　确认“退出 WinCC 运行系统”

## 4.4.3　变量连接

将对象的某个物理属性与变量进行连接，变量的值将直接赋给该对象的对应属性，使其动态化。

**1. 变量连接的组态方法**

选中组态变量连接的对象，并打开“对象属性”窗口，在“属性”选项卡中选择想要动态化的属性，用鼠标右键单击“动态”列上的白色灯泡，在弹出的快捷菜单中选择“变量…”命令，打开“变量 - 项目”窗口中选择需要连接的变量。完成后，“动态”列上的绿色灯泡和连接的变量名称说明了改属性已经组态了变量连接的动态化。“更新周期”列如果没有进行修改，图形编辑器中的默认触发器设置将作为其更新周期。

**2. 组态实例**

（1）画面对象的显示属性

在组态动态化时，经常遇到更改某个画面对象的显示/隐藏状态的问题，这需要针对画面对象的“显示”属性进行组态。画面对象的“显示”属性可以和一个 BOOL 型的

WinCC 变量连接，当 BOOL 变量的值为 1 时，该画面对象显示，值为 0 时，画面对象隐藏。以下以汽车是否到达喷漆位置为例（只有到达喷漆位置才显示），用变量连接的方法实现画面对象的显示/隐藏状态更改的组态。

1）单击图形编辑器菜单栏“视图”→“库”，或单击工具栏的按钮，打开图形编辑器的库，在“全局库”→“Siemens HMI Symbol Library 1.3”→“交通工具”中，找到“汽车 1”，并将其添加到画面的工作区。

2）用鼠标右键单击添加的画面对象，在弹出的快捷菜单中单击“属性”，打开“对象属性”窗口。

3）激活“属相”选项卡，并在属性列表中选择“其他”→“显示”属性，用鼠标右键单击“动态”列上的白色灯泡，在弹出的快捷菜单中选择“变量…”，打开“变量-项目”窗口选择连接的变量“到达喷漆位置”，如图 4-27 所示。组态变量连接后，白色灯泡变为绿色灯泡。

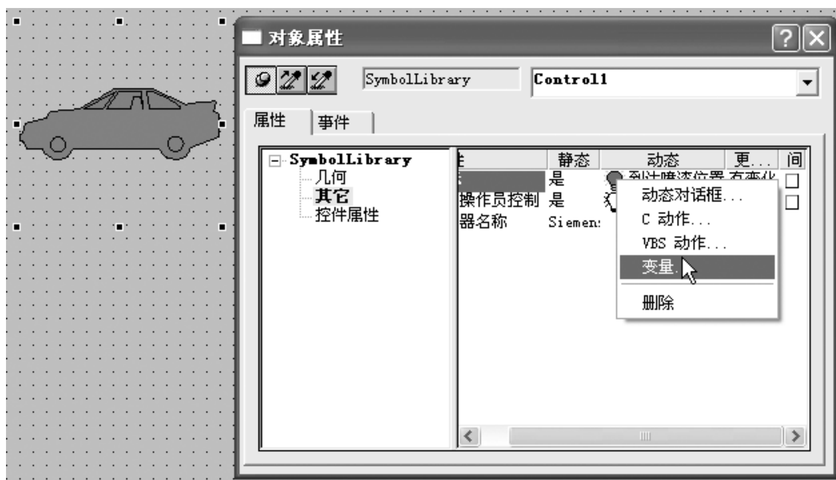



图 4-27　应用“变量连接”组态对象的“显示”属性

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

（2）填充量

为了演示应用“变量连接”使对象属性“填充量”动态化，在图形编辑器中添加一个圆角矩形和一个滚动条对象，在 WinCC 变量管理器中创建一个内部变量，组态圆角矩形的“填充量”和滚动条的“过程驱动连接”属性与该内部变量连接，详细步骤如下：

1）在变量管理器的内部变量文件夹中新建“FillLevel”变量，数据类型为“无符号 8 位数”（因填充量的取值范围是 0～100）。

2）激活图形编辑器的对象选项板中的“标准”选项卡，选择“标准对象”→“圆角矩形”，并将其添加到画面的工作区。

3）用鼠标右键单击圆角矩形，在弹出的快捷菜单中选择“属性”，打开“对象属性”窗口。激活“属性”选项卡，在左侧窗口的属性列中选择“填充”，在右侧窗口将“动态填充”属性的静态值设置为“是”，“填充量”属性的静态值设置为 30，用鼠标右键单击“填充量”属性的“动态”列上的白色灯泡，在弹出的快捷菜单中选择“变量



…”，打开“变量 - 项目”窗口，选择连接的变量“FillLevel”，如图 4-28 所示。




图 4-28　应用“变量连接”组态“填充量”属性

4）在对象选项板中的“标准”选项卡中选择“窗口对象”→“滚动条”，并将其添加到画面的工作区，将自动弹出“滚动条组态”对话框，如图 4-29 所示，单击按钮，打开“变量 - 项目”窗口，选择连接的变量“FillLevel”。按实际需要设置“限制”框中的“最大值”、“最小值”和“步长”等参数。

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，用鼠标拖动画面中的滚动条，观察圆角矩形填充量的变化，测试组态的结果。




图 4-29　设置“滚动条”的组态对话框

## 4.4.4　动态对话框

如果实现对象的属性动态变化需要执行范围选择、状态判断、函数运算等复杂的控制，则可以借助动态对话框的方法来组态。实际上，动态对话框是一个简化的脚本编程，根据用户输入的信息将其转化为 C 脚本程序。

**1. 动态对话框组态窗口**

在“对象属性”窗口中选择“属性”选项卡，在右侧窗口用鼠标右键单击需要动态化的属性的白色灯泡，在弹出的快捷菜单中选择“动态对话框…”，将打开动态对话框的组态窗口，如图 4-30 所示，其选项和参数设置说明如下：

（1）事件名称

“事件名称”框用于设置动态对话框所组态的动态的触发器，如果没有设置触发器，则由系统指定触发事件的默认值，默认值取决于动态对话框中“表达式/公式”的内容。单击按钮，打开“改变触发器”对话框，如图 4-31 所示。在“事件”区域选择触发器

的类型，可以是变量触发器，也可以是基于时间触发器类型的标准周期、画面周期或窗口周期。在“触发器名称”框可以为组态的触发器指定一个名称，而“周期”框设定的是时间触发器的更新周期。




图 4-30　“动态对话框”组态窗口




图 4-31　设置触发器

（2）表达式/公式

指定用于定义对象属性的表达式，表达式可以是一个变量这样的简单表达式，也可以是包含算术运算（加 +、减 -、乘 *、除/）和逻辑运算（与 &&、或 | |、非!）的复杂表达式，还可以是 C 脚本函数。可以直接在框中输入表达式，也可以单击...将变量、C 脚本函数和操作符添加到表达式中。单击“检查”按钮，可以对表达式的语法进行检查。

（3）表达式/公式的结果

根据表达式/公式的结果，进行范围选择和状态判断等，从而定义对象属性不同的属性值。表达式/公式的结果还与“数据类型”的设置相关。

（4）数据类型

有 4 种数据类型可供选择，分别如下：

- 模拟量：可以单击“添加…”按钮添加模拟量限制值内的多个数值范围，对象属性的值随数值范围变化而变化。
- 布尔型：用“真”（1）和“假”（0）定义两种不同的状态，对象属性的值也因此有两个不同的值。
- 位：定义某个字节（或字或双字）变量的一个位，其值（0/1）决定了对象属性的值。
- 直接：将“表达式/公式”的值用做对象属性的值（与“变量连接”不同的是只能用于输出量）。

（5）变量状态

用于监视运行系统中 WinCC 变量的状态。

（6）质量代码



用于监视运行系统中 WinCC 变量的质量代码。

2. 组态实例

以下根据图 4-30 所示“数据类型”框中提供的 4 种选择介绍各自的组态实例。

（1）模拟量

以本章 4.4.1 小节的图 4-20 中组态的棒图为例，介绍棒图的颜色随模拟量的取值范围不同发生变化的组态方法。具体表现为：变量“温度”≤25 时棒图颜色为绿色，25＜“温度”≤40 为黄色，“温度”＞40 为红色。组态方法如下：

1）用鼠标右键单击棒图，在弹出的快捷菜单中选择“属性”，打开“对象属性”窗口。激活“属性”选项卡，在左侧窗口的属相列中选择“颜色”属性，右侧窗口中用鼠标右键单击“棒图颜色”属性“动态”列上的白色灯泡，在快捷菜单中选择“动态对话框”，打开“动态值范围”窗口，如图 4-32 所示。

2）在窗口中选择数据类型为“模拟量”，单击…按钮，在弹出的快捷菜单中选择“变量”，打开“变量-项目”窗口连接变量“温度”。单击按钮，打开“改变触发器”窗口，设置触发器类型为变量触发。单击“添加（A）…”按钮，添加变量“温度”的取值范围。如图 4-32 所示，在“表达式/公式的结果”栏中组态棒图的颜色，“数值范围 1”对应的是变量“温度”≤25，棒图颜色为绿色；“数值范围 2”对应的是 25＜“温度”≤40，棒图颜色为黄色；“其他”对应的是“温度”＞40，棒图颜色为红色。单击“应用”按钮，关闭“动态值范围”窗口。




图 4-32　应用“动态对话框”组态棒图颜色

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

（2）布尔型

在 HMI 系统中，通常都要求可以非常直观地查看监控对象的运行状态，这可以通过在过程画面中组态指示灯来实现。例如，对于一条受监控的生产线，当过程画面的指示灯为绿色时，表示生产线处于运行状态，指示灯为红色时，生产线处于停机状态，这与生产线现场的信号灯是一致的。假定二进制类型的 WinCC 过程变量“生产线运行”反映生产线的运行状态，以下为在过程画面组态反映生产线运行状态的指示灯的组态步骤：

1）在画面中添加一个圆作为指示灯，用鼠标右键单击圆，在弹出的快捷菜单中选择“属性”，打开“对象属性”窗口。激活“属性”选项卡，在左侧窗口的属性列中选择

“颜色”属性，右侧窗口中用鼠标右键单击“背景颜色”属性“动态”列上的白色灯泡，在快捷菜单中选择“动态对话框”，打开“动态值范围”窗口，如图 4-33 所示。

2）在窗口中选择数据类型为“布尔型”，单击按钮，在弹出的快捷菜单中选择“变量”，打开“变量－项目”窗口连接变量“生产线运行”。单击按钮，打开“改变触发器”窗口，设置触发器类型为变量触发。如图 4-33 所示，在“表达式/公式的结果”栏中组态指示灯的颜色，“是/真”对应的是变量“生产线运行”值为 1，颜色为绿色；“否/假”对应的是变量“生产线运行”值为 0，指示灯颜色为红色。单击“应用”按钮，关闭“动态值范围”窗口。




图 4-33　应用“组态对话框”组态指示灯

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

（3）位

假定 HMI 监控的生产线有 8 个工位，每个工位对应一个指示灯。当工件到达某个工位时，相应的指示灯闪烁。如果采用“变量连接”的方法，为这 8 个指示灯连接 8 个二进制的变量，将占用 8 个 WinCC Power Tags 授权。应用“动态对话框”的“位”数据功能，可以用一个字节变量（或字变量、双字变量）代替 8 个二进制的变量，而占用的 Power Tags 授权仅为 1 个。组态步骤如下：

1）在 WinCC 变量管理器中新建一个名称为“工位指示灯”的变量，数据类型为“无符号 8 位数”，对应的地址为 MB90。

2）在图形编辑器中添加 8 个圆作为“工位指示灯”到画面的工作区，组态每个圆的闪烁属性，连接“工位指示灯”变量的对应位。用鼠标右键单击 1 号工位的指示灯，在弹出的快捷菜单中选择“属性”，打开“对象属性”窗口，激活“属性”选项卡，如图 4-34 所示。在左侧窗口的属性列中选择“闪烁”属性，在右侧窗口中用鼠标右键单击“闪烁背景激活”属性“动态”列上的白色灯泡，在弹出的快捷菜单中选择“动态对话框”命令，打开“动态值范围”窗口。

3）如图 4-35 所示，在数据类型框中选择“位”，单击变量选择按钮（左侧），在打开的“变量－项目”窗口中连接变量“工位指示灯。单击按钮，打开“改变触发器”窗口，设置触发器类型为变量触发。单击位地址选择按钮（右侧），在弹出的“位选择”对话框中第 0 位。在“表达式/公式结果”区域设置“闪烁背景激活”属性，



“置位”表示 M90.0 为 1，激活 1 号工位指示灯背景闪烁；“未置位”表示 M90.0 为 0，不激活 1 号工位指示灯背景闪烁。单击“应用”按钮，关闭“动态值范围”窗口。

4）在“对象属性”窗口的左侧窗口选择“闪烁”，在右侧窗口中设置“闪烁背景颜色开”为绿色，设置“闪烁背景颜色关”为白色，设置“背景闪烁频率”为中等。

5）应用同样的方法组态 2 号工位至 8 号工位指示灯，“闪烁背景激活”属性分别连接到“工位指示灯”变量 MB90 的第 1 位 M90.1 至第 7 位 M90.7。

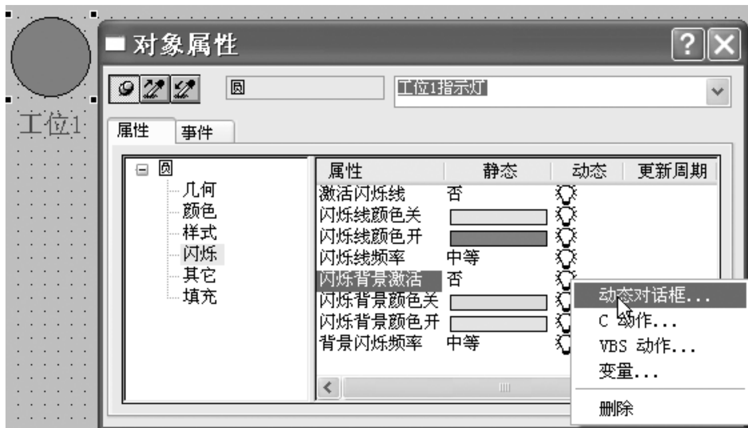



图 4-34　组态工位指示灯的“闪烁背景激活”属性




图 4-35　组态工位指示灯闪烁属性的位地址

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

（4）直接

如图 4-36 所示，假定液体罐的容量为 1000 升（L），在液体罐上开矩形窗口，以矩形窗口的填充量多少反映液体罐的实际液位（范围 0～1000L）。由于矩形的填充量是按百分比计算（范围 0～100，图 4-36 中的 60 即 60%），因此不能直接将反映液体罐液位的变量与矩形的“填充量”属性做变量连接，需要进行换算。可以借助“动态对话框”的“直接”数据类型实现，组态方法如下：

1）在 WinCC 变量管理器中新建一个名称为“实际液位值”的变量，数据类型为“无符号 16 位数”，设置上限为 1000，下限为 0。

2）用鼠标右键单击图 4-36 中的矩形，在弹出的快捷菜单中选择“属性”，打开“对象属性“对话框”，激活“属性”选项卡。在左侧窗口的属性列中选择“填充”属性，在右侧窗口中设置“动态填充”属性的静态值为“是”，“填充量”属性的静态值为 60，用鼠标右键单击“填充量”属性“动态”列上的白色灯泡，在弹出的快捷菜单中选择“动态对话框”命令，打开“动态值范围”窗口。

3）如图在窗口中选择数据类型为“直接”，单击⋯按钮，在弹出的快捷菜单中选择“变量”，打开“变量 - 项目”窗口连接变量“实际液位值”。单击按钮，打开“改变触发器”窗口，设置触发器类型为变量触发。在“表达式/公式”栏中输入计算公式：“实际液位值”/10（在已经添加的变量“实际液位值”后输入“/10”即可）。单击“应用”按钮，关闭“动态值范围”窗口。




图 4-36　组态“矩形”的“填充量”属性




图 4-37　“直接”设置矩形的“填充量”属性

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

注：图 4-36 中的液体罐、阀和管道均取自于图形编辑器的图形库。



### 4.4.5 直接连接

直接连接用于组态对象的事件动作。直接连接在画面中提供最快速的动态并可获得最好的运行性能。然而直接连接只能用在过程画面中，并且只能创建一个连接。

**1. 直接连接的组态窗口**

打开“对象属性”窗口后，选择“事件”选项卡，在左侧窗口的列表中选择事件组（例如“鼠标”事件组），在右侧窗口用鼠标右键单击要组态动作的事件的“动作”列，在弹出的快捷菜单中选择“直接连接”，打开直接连接的组态窗口，如图 4-38 所示。

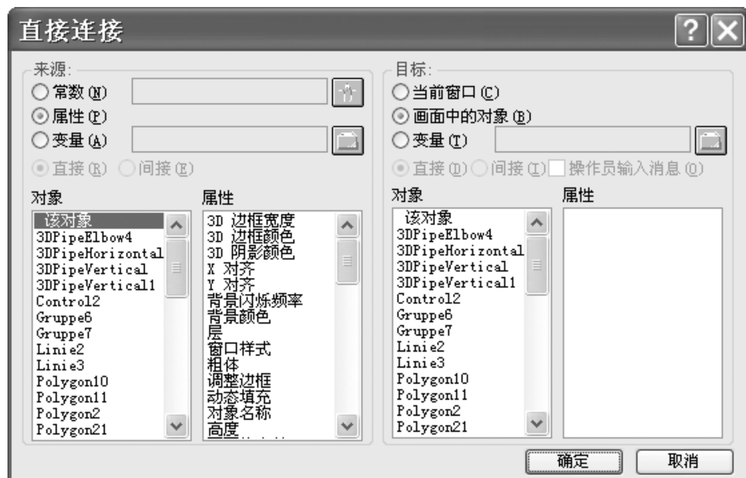



图 4-38　直接连接的组态窗口

直接连接组态窗口的左侧是“来源”框，右侧是“目标”框，直接连接组态的事件动作是一个赋值操作，即：“目标” = “来源”。“来源”可以是常数、对象的属性和变量等源元素，目标可以是当前窗口（过程画面）的属性、画面对象的属性和变量等目标元素。当事件在运行系统中发生时，其对应的时间动作将“来源”的元素值赋给“目标”元素。可以看出，直接连接与变量连接相比，具有应用范围更广泛的赋值动态。

**2. 组态实例**

直接连接多数情况下用于组态对现场进行控制的动态，例如设置现场设备的运行参数，远程启动/停止设备的运行等。

（1）组态远程启动/停止按钮

一般情况下，组态了 HMI 的自动化系统存在本地控制和远程控制两种方式。本地控制方式即通过系统现场的控制面板进行控制，对设备进行维修和保养时一般采用这种方式。远程控制即通过 HMI 对系统进行控制，当系统正常运行和生产的时候采用。为了避免本地控制和远程控制同时操作可能产生的不安全因素，在现场的控制面板上要设置一个“本地/远程”控制的选择开关，实现“本地/远程”控制的互锁。

以下为组态远程启动/停止按钮的方法：

1）在 WinCC 变量管理器中创建名称为“远程启动”和“远程停止”的二进制变量，地址分别为 M1.0、M1.1。

2）在图形编辑的对象选项板中，选择“窗口对象”→“按钮”，添加 2 个按钮到画面的工作区，并将其显示文本分别修改为“启动”、“停止”。

3）用鼠标右键单击“启动”按钮，打开“对象属性”对话框，如图 4-39 所示。激活“事件”选项卡，在左侧窗口选择“鼠标”，在右侧窗口用鼠标右键单击“按左键”事件“动作”列上的白色闪电，在快捷菜单中选择“直接连接”，弹出“直接连接”窗口。

4）如图 4-40 所示，在“来源”框中选择“常数”，并输入常数 1。在“目标”框中选择“变量”，单击按钮，打开“变量 – 项目”窗口，连接变量“远程启动”。单击“确定”按钮，关闭“直接连接”窗口。

5）用鼠标右键单击“释放左键”事件“动作”列的白色闪电，在弹出的快捷菜单中选择“直接连接”，打开“直接连接”窗口，按步骤 4）介绍的方法将左侧“来源”的常数 0 与右侧“目标”的变量“远程启动”直接连接。

6）用同样的方法组态“停止”按钮，按鼠标左键时将常数 1 与变量“远程停止”直接连接，释放左键时将常数 0 与变量“远程停止”直接连接。




图 4-39　应用直接连接组态“启动”按钮的事件动作

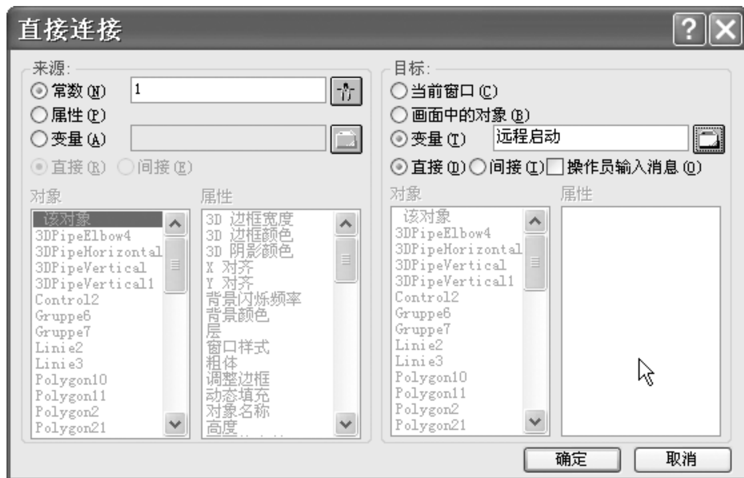



图 4-40　“直接连接”窗口组态“启动”按钮的动作



保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

（2）组态画中画显示

在实际应用中，对于一些过程画面中的监控对象，例如电机、电磁阀和泵等，当需要监控的参数较多时，可以考虑采用画中画显示。也就是将监控这些对象需要的画面元素组态到一个单独的画面中，一般情况下隐藏在监控对象所在的过程画面中，当需要查看或设置其参数时，单击相应的画面元素，打开为其单独组态的画面。

如图 4-41 所示，单击电机下方带向右箭头的按钮，将打开电动机的监控画面，单击电动机监控画面的“关闭窗口”按钮，将关闭画中画显示，下面将介绍用直接连接的方法组态画中画的显示与隐藏。

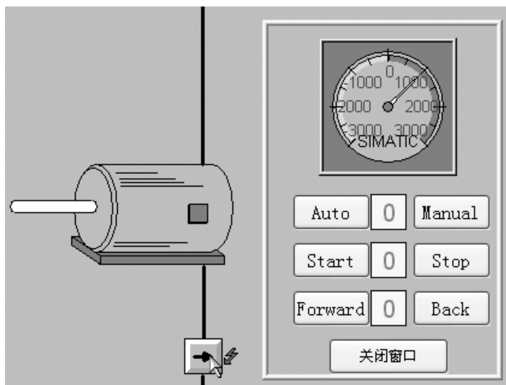



图 4-41　画中画显示实例

1）组态电机的监控画面。在图形编辑器中新建画面，用鼠标右键单击画面，打开“对象属性”窗口，在“属性”选项卡中选择“几何”属性，在右侧窗口中设置画面宽度为 200，高度为 300。在画面的下方添加一个按钮，文本为“关闭窗口”。用鼠标右键单击该按钮，打开“对象属性”窗口，如图 4-42 所示。在“事件”选项卡中选择“鼠标”，在右侧窗口中用鼠标右键单击“按左键”事件“动作”列上的白色闪电，在弹出的快捷菜单中选择“直接连接”，打开“直接连接”组态窗口，如图 4-43 所示。左侧“来源”选择常数 0，右侧“目标”选择“当前窗口”的“显示”属性进行直接连接。电机监控画面其他画面对象的添加与动态组态这里不详述。保存画面，画面名称为“Motor. Pdl”。

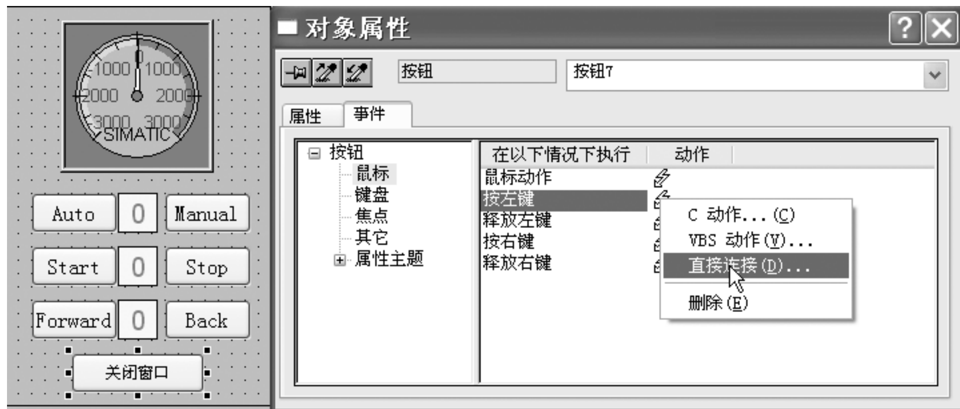



图 4-42　应用直接连接组态“关闭窗口”按钮

2）组态画面窗口。图 4-41 中的“电机”和带向右箭头的“按钮”均取自与图形编辑器的图形库。选择“对象选项板”→“智能对象”“画面窗口”，将其添加到画面的工作区。用鼠标右键单击“画面窗口”，在弹出的快捷菜单中选择“属性”，打开“对象属性”窗口。在“属性”选项卡中选择“几何”属性，设置“画面窗口”的宽度为 200，高度为 300。在左侧窗口选择“其他”属性，在右侧窗口设置“显示”属性的静态值为“否”，“边框”属性的静态值为“是”，“画面名称”属性的静态值为“Motor. Pdl”，如图 4-44 所示。




图 4-43　“直接连接”窗口组态“关闭窗口”按钮的动作




图 4-44　组态“画面窗口”对象的属性

3）组态显示画中画的按钮。用鼠标右键单击图 4-41 中带向右箭头的按钮，打开“对象属性”窗口。在“事件”选项卡中选择“鼠标”，在右侧窗口中用鼠标右键单击“按左键”事件“动作”列上的白色闪电，在弹出的快捷菜单中选择“直接连接”，打开组态窗口。左侧“来源”选择常数 1，右侧“目标”选择“画面中的对象”→“画面窗



口 1”的“显示”属性进行直接连接，如图 4-45 所示。

4）保存电机所在的画面，在图形编辑器或项目管理器中激活项目，测试画中画显示的组态结果。




图 4-45 “直接连接”窗口组态显示画中画的按钮

## 4.4.6 C 动作

C 动作可用于组态对象的属性动态化和事件动作。当组态对象属性动态化时，对象属性的值将由 C 函数的返回值来确定，且必须组态属性动态的触发器。作用于对象的事件动作由对象属性变化的事件或其他事件来激活。

WinCC 的 C 脚本语言基于 ANSI C 标准，并允许用最大的灵活性定义动态对象。如果变量连接、动态对话框和直接连接等方法不能解决对象的动态化问题，则可采用 C 动作实现。

注意：编写 C 动作程序时，要注意 C 脚本的数据类型，请参考第 3 章 3.2.2 小节的表 3-2。

### 1. C 动作编辑器

在“对象属性”窗口，选择“属性”选项卡时，用鼠标右键单击对象属性的“动态”列的白色灯泡；选择“事件”选项卡时，用鼠标右键单击对象事件的“动作”列的白色闪电，在各自弹出的快捷菜单中选择“C 动作”命令，都可以打开 C 动作编辑器，如图 4-46 所示。

工具栏几个常用的按钮的功能如下：

- 按钮：用于编译 C 动作函数。该过程将由消息“编译动作 ...”在对话框的状动态中进行说明。如果没有错误地完成编译，则消息“0 Error，0 Warning”将显示在状态栏中。每次修改后都要对 C 动作进行编译。
- 按钮：用于打开“变量 – 项目”窗口，连接 C 动作中需要访问的 WinCC 变量。
- 按钮：用于打开“画面”窗口，选择组态 C 动作时需要访问的过程画面。
- 按钮：用于导入 C 动作。
- 按钮：用于导出 C 动作。

- 按钮：用于打开“改变触发器”对话框，组态对象属性动态化的触发器。当组态对象的事件动作时，该按钮无法操作。

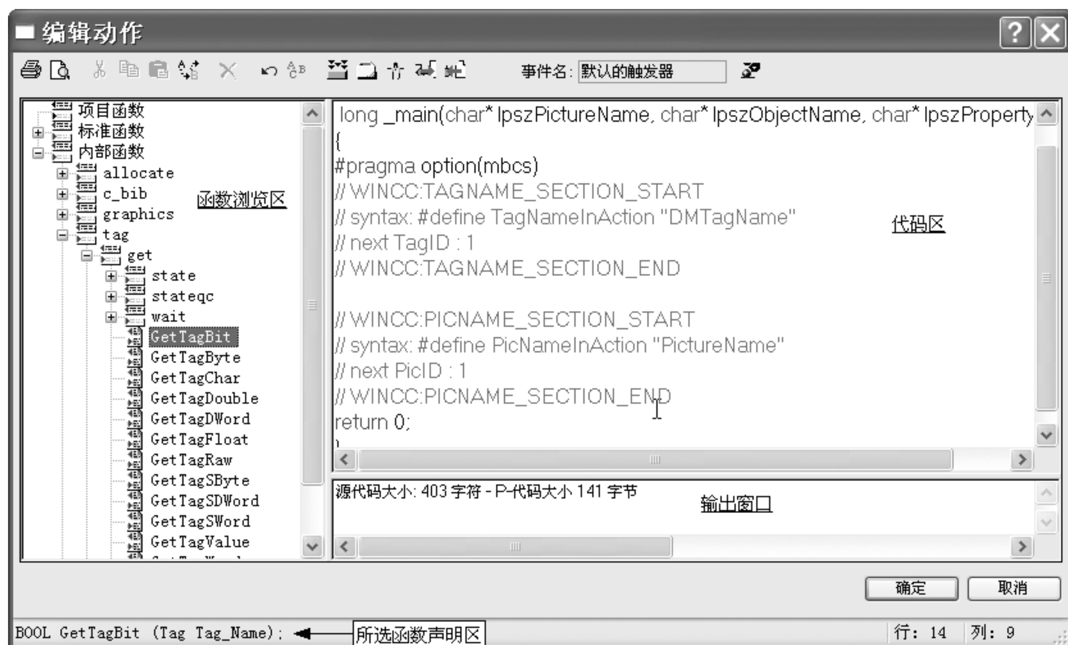



图 4-46　C 动作编辑器

## 2. 组态实例

(1) C 动作组态对象的属性动态化

在实际应用中，经常需要组态一些动画，例如，对象移动的动画、旋转角度的动画等。组态这些动画最常用的方法就是用 C 动作。

如图 4-47 所示，画面中的实线模拟时钟的秒针旋转，每隔 1 秒旋转 6 度，I/O 域中显示了实线当前的旋转角度（0 度时直线与图形编辑器的 Y 轴正方向一致），组态方法如下：

1）在变量管理器中创建名称为“angel1”的内部变量，数据类型为无符号 16 位数。

2）在图形编辑器的对象选项板中选择“智能对象”→“输入/输出域”，添加到画面的工作区，并在弹出的“I/O 域组态”对话框中将“输入/输出域”与变量“angel1”连接。

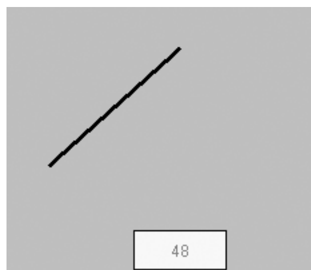



图 4-47　对象旋转动画实例

3）在图形编辑器的对象选项板中选择“标准对象”→“线”，添加到画面的工作区。如图 4-48 所示，用鼠标右键单击“线”，在弹出的快捷菜单中选择“属性”，打开“对象属性”窗口。在“属性”选项卡中选择“几何”属性，在右侧窗口中用鼠标右键单击“旋转角”属性“动态”列上的白色灯泡，在弹出的快捷菜单中选择“C 动作…”，打开 C 动作编辑器。

4）单击按钮，将 C 动作的触发器设置为基于标准周期的时间触发，周期为 1 秒。在代码区输入如下程序：



```
{
    short int back;                       //定义一个短整型变量
    back = GetTagWord("angel1");          //获取字变量"angel1"的当前值(角度)
    back = back + 6;                      //将角度值增加 6 度
    SetTagWord("angel1",back%360);        //将增加后的角度值赋给变量"angel1"
    return back;                          //将角度值返回给"旋转角"属性
}
```

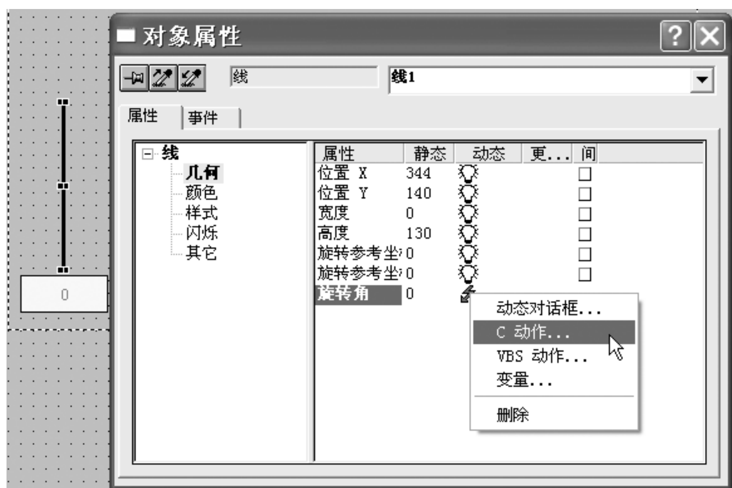



图 4-48　组态对象属性的 C 动作

5）单击按钮，编译动作，如有错误提示，修改错误后重新编译，直至没有错误提示，单击“确定”按钮，完成 C 动作的编辑退出 C 动作编辑器。

程序中 WinCC 内部函数 GetTagWord 的应用如图 4-49 所示，在“函数浏览区”选择“内部函数”→“tag”→“get”→“GetTagWord”，在弹出的“分配参数”对话框中单击按钮，在快捷菜单中选择“变量选择”命令，打开“变量 - 项目”窗口，选择变量“angel1”。

内部函数 SetTagWord 的应用如图 4-50 所示，在“函数浏览区”选择“内部函数”→“tag”→“set”→“SetTagWord”，弹出的“分配参数”对话框。在“Tag_Name”行，单击按钮，在快捷菜单中选择“变量选择”命令，打开“变量 - 项目”窗口，选择变量“angel1”。在“value”行，键入表达式“back%360”（该表达式将变量“angel1”的取值范围限定到 0 ~ 360）。

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

（2）C 动作组态对象的事件动作

在过程画面中组态一个按钮，实现开关功能，解决远程控制电机的启动/停止、阀门的打开/关闭等问题。

应用直接连接的方法解决类似问题时，需要组态两个按钮（参考 4.4.5 小节的“组态实例”→“（1）组态远程启动/停止按钮”）。用 C 动作实现只需要组态一个按钮，具体组态方法如下：

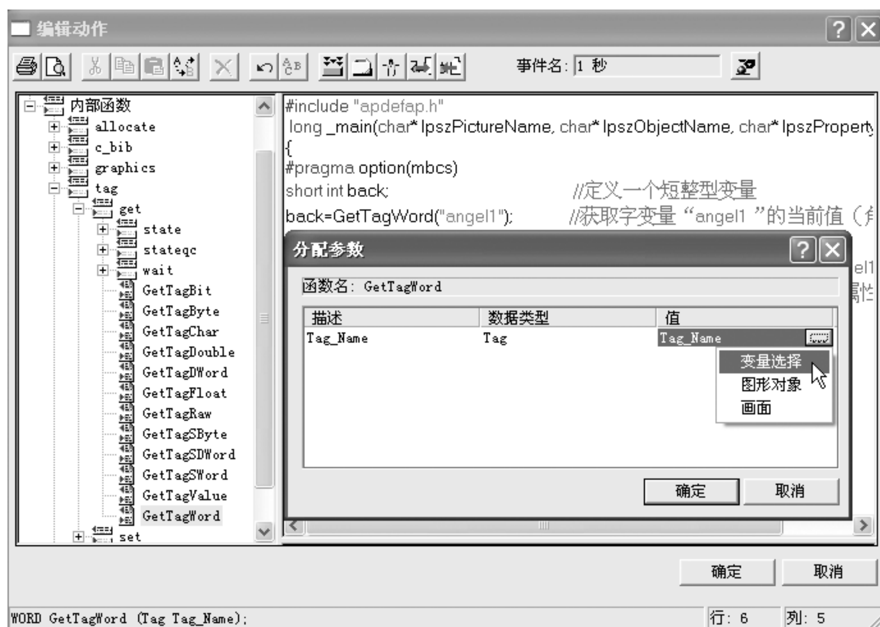



图 4-49　内部函数 GetTagWord 的应用

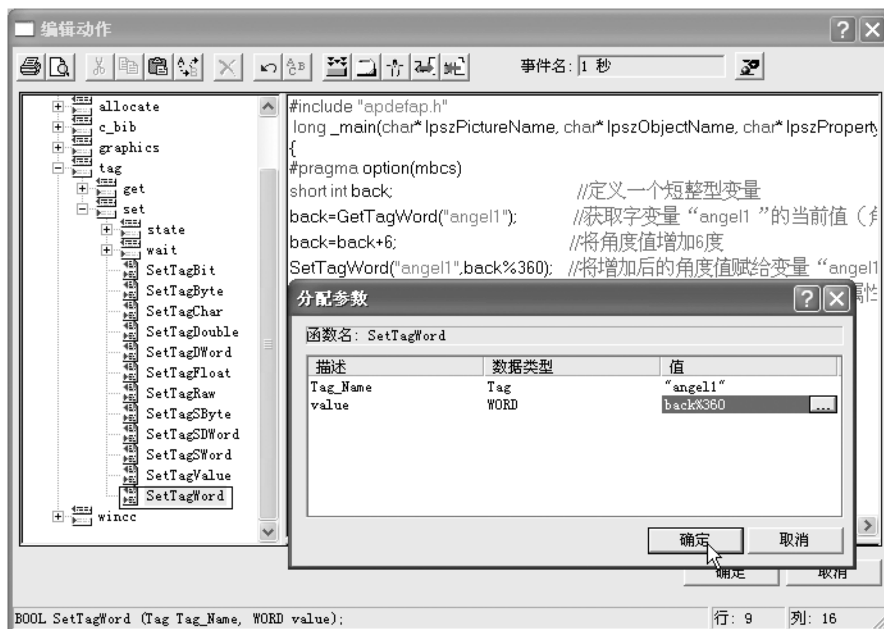



图 4-50　内部函数 SetTagWord 的应用

1）在 WinCC 变量管理器中新建一个名称为“电机开关”的二进制变量，地址为 M1.2。

2）添加一个按钮到过程画面的工作区，文本显示为“电机开关”。用鼠标右键单击按钮，打开“对象属性”对话框，如图 4-51 所示。激活“事件”选项卡，在左侧窗口选择“鼠标”，在右侧窗口用鼠标右键单击“按左键”事件“动作”列上的白色闪电，在快捷菜单中选择“C 动作…”，打开 C 动作编辑器。在代码区输入如下程序：



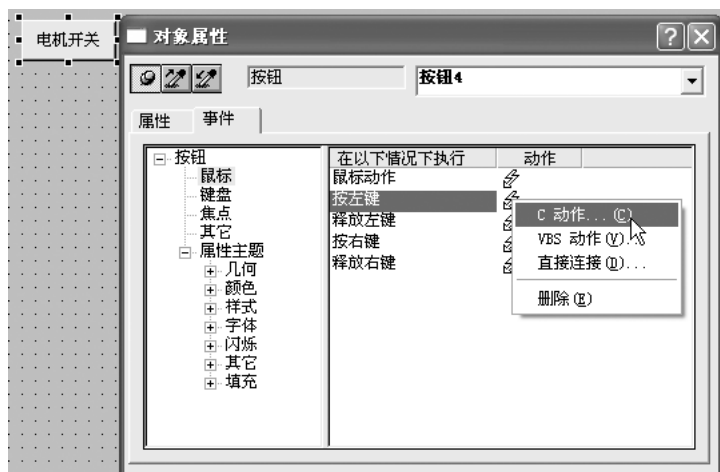



图 4-51　应用“C 动作”组态按钮的事件动作

```
{
    BOOL  temp;                         //定义一个 BOOL 型变量
    temp = GetTagBit("电机开关");        //读取变量“电机开关”的当前值
    temp = ! temp;                      //将变量当前值取反
    SetTagBit("电机开关",temp);          //将取反后的值赋值给变量“电机开关”
}
```

3）在按钮的“对象属性”窗口激活“属性”选项卡，在左侧窗口的属性列表中选择“字体”，在右侧窗口用鼠标右键单击“文本”属性“动态”列上的白色灯泡，在弹出的快捷菜单中选择“动态对话框”命令，打开“动态值范围”窗口，如图 4-52 所示。




图 4-52　应用“动态对话框”组态按钮的“文本”属性

4）在窗口中选择数据类型为“布尔型”，单击按钮，在弹出的快捷菜单中选择“变量”，打开“变量 - 项目”窗口连接变量“单击开关”。单击按钮，打开“改变触发器”窗口，设置触发器类型为变量触发。在“表达式/公式的结果”栏中组态按钮的文本，“是/真”对应的是变量“电机开关”值为 1，当前电机处于启动状态，单击该按钮的动作应该是停止电机运行，因此显示文本为“停止电机”；“否/假”对应的是变量“电机开关”值为 0，当前电机处于停止状态，单击该按钮的动作应该是启动电机，因此显示文本为“启动电机”。单击“应用”按钮，关闭“动态值范围”窗口。

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

## 4.4.7　VBS 动作

从 WinCC V6.0 开始，引入了 VBScript，用于组态对象的属性动态化和事件动作，用户多了一种组态方法的选择。如果变量连接、动态对话框和直接连接等方法不能解决对象的动态化问题，除可以采用 C 动作实现外，也可以采用 VBS 动作实现。

### 1. VBScript 的相关概念

（1）VBS 的数据类型

VBS 使用 Variant 数据类型，Variant 数据类型自动适应所赋数值的类型。编写 VBS 动作程序时，声明变量或对象的基本语法如下，无需指定数据类型：

```
Dim obj1
```

这里 obj1 可以是 WinCC 变量，也可以是 WinCC 画面或画面对象，由为其赋值的数值的类型确定。

（2）VBScript 的对象参考模型

WinCC 的 VBS 也有对象、属性、方法的概念。要用 VBS 动作针对 WinCC 的变量、画面和画面对象等组态动态化，必须清楚 WinCC 的 VBScript 对象模型。WinCC 的 VBScript 对象模型如图 4-53 所示，对象模型类似于一种树形结构，最顶层对象（根对象）为 HMIRuntime（即 HMI 运行系统）对象，下一层对象为根对象的子对象，依次类推。组态过程画面动态化时访问最多的对象是 WinCC 变量（“Tags”→“Tag”）和画面对象（“ScreenItems”→“ScreenItem”）。

访问 WinCC 变量的方法如下：

```
Dim my_Soll1        ‘声明一个对象
Set my_Soll1 = HMIRuntime.Tags（“Soll1”）        ‘将声明的对象指向要访问的 WinCC 变量
```

访问画面对象的方法如下：

```
Dim Rect            ‘声明一个对象
Set Rect = ScreenItems（“RoundedRectangle1”）  ‘将声明的对象指向要访问的画面对象
```

将声明的对象指向要访问的 WinCC 变量或画面对象后，就可以通过操作这个声明的对象来实现对 WinCC 变量或画面对象的操作。

（3）VBS 动作编辑器

在“对象属性”窗口中，选择“属性”选项卡时，用鼠标右键单击对象属性的“动



态”列的白色灯泡；选择“事件”选项卡时，右键单击对象事件的“动作”列的白色闪电，在各自弹出的快捷菜单中选择“VBS 动作”命令，都可以打开 VBS 动作编辑器，如图 4-54 所示。

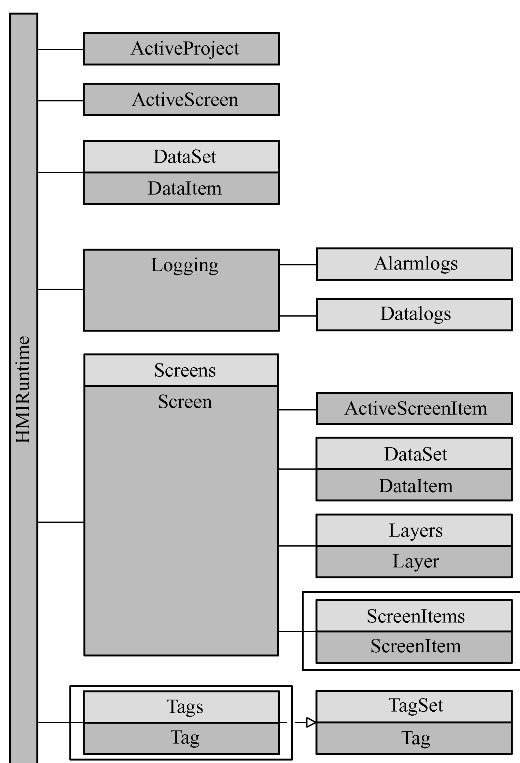



图 4-53　WinCC 的 VBScript 对象模型

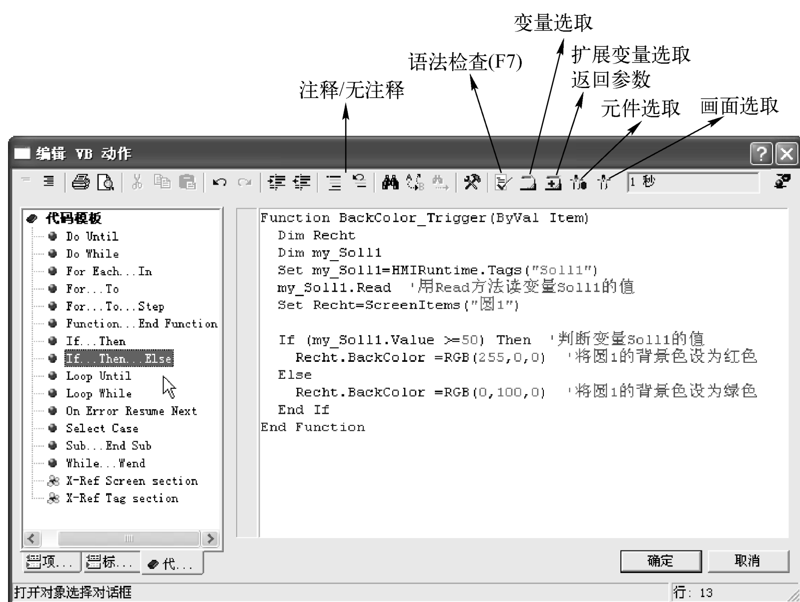



图 4-54　VBS 动作编辑器

注意：VBS 动作编辑器提供智能工具，HMIRuntime、ScreenItems 等关键字可以通过智能工具自动添加到代码区。如图 4-55 所示，在代码区单击鼠标右键，在弹出的快捷菜单中选择“对象列表”、“属性/方法”、“功能列表”和“快速指导信息”等命令，可以在弹出的快捷菜单中选择自动添加的关键字。

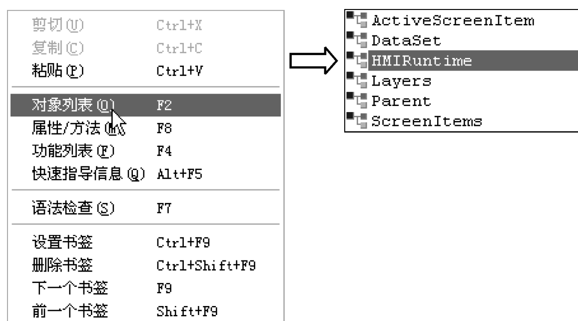



图 4-55　VBS 动作编辑器智能工具

### 2. 组态实例

（1）组态退出 WinCC 运行系统的按钮——事件动作

本章 4.4.2 节介绍了用动态向导方法组态退出 WinCC 运行系统的按钮，动态向导将自动生成 C 脚本程序。这里介绍用 VBScript 组态具有退出 WinCC 运行系统功能的按钮，步骤如下：

1）在图形编辑器的对象选项板中，选择“窗口对象”→“按钮”，添加 1 个按钮到画面的键集合区，并将其显示文本修改为“退出 WinCC 运行系统”。

2）用鼠标右键单击该按钮，打开“对象属性”对话框。激活“事件”选项卡，在左侧窗口选择“鼠标”，在右侧窗口用鼠标右键单击“按左键”事件“动作”列上的白色闪电，在快捷菜单中选择“VBS 动作…”，打开 VBS 动作编辑器。在代码区输入如下程序：

```
Sub OnLButtonDown (ByVal Item, ByVal Flags, ByVal x, ByVal y)
  HMIRuntime.Stop
End Sub
```

3）单击“语法检查”按钮，如有错误提示，修改错误后重新进行检查，直至没有错误提示，单击“确定”按钮，完成 VBS 动作的编辑退出 VBS 动作编辑器。

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

（2）定义运行画面的样式和颜色——属性动态化

在 WinCC 运行状态下，打开过程画面，该画面按 VBS 脚本程序中定义的样式和颜色显示。组态步骤如下：

1）打开需要组态的画面，用鼠标右键单击该画面，打开“对象属性”对话框。激活“事件”选项卡，在左侧窗口选择“其他”，在右侧窗口用鼠标右键单击“打开画面”事件“动作”列上的白色闪电，在快捷菜单中选择“VBS 动作…”，打开 VBS 动作编辑器，如图 4-56 所示。在代码区输入如下程序：

```
Sub OnOpen()
```



```
    Dim objScreen
    Set objScreen = HMIRuntime. Screens("运行画面")
    objScreen. FillStyle  = 131075    设置画面的填充模式
    objScreen. FillColor  = RGB(0,0,255)    设置画面的填充颜色
End Sub
```

2）单击“语法检查”按钮，如有错误提示，修改错误后重新进行检查，直至没有错误提示，单击“确定”按钮，完成 VBS 动作的编辑退出 VBS 动作编辑器。

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

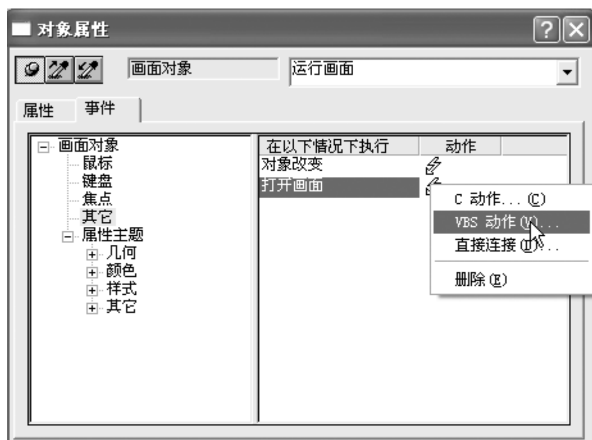



图 4-56　组态“打开画面”事件的 VBS 动作

（3）单击按钮显示/隐藏一个帮助信息——事件动作

帮助信息加载到一个静态文本对象中，单击按钮切换静态文本在过程画面中的显示和隐藏。组态步骤如下：

1）在图形编辑器的对象选项板中，选择“窗口对象”→“按钮”，添加 1 个按钮到画面的工作区，并将其显示文本修改为“帮助”。

2）在图形编辑器的对象选项板中，选择“标准对象”→“静态文本”，添加 1 个静态文本到画面的工作区。用鼠标右键单击该静态文本，打开“对象属性”对话框，激活“属性”选项卡，在左侧窗口选择“字体”属性，双击右侧窗口“文本”属性的“静态”列，在编辑框中输入需要显示的帮助信息。在左侧窗口选择“其他”属性，双击右侧窗口“显示”属性的“静态”列，将“显示“属性的静态值修改为“否”。

3）用鼠标右键单击该按钮，打开“对象属性”对话框。激活“事件”选项卡，在左侧窗口选择“鼠标”，在右侧窗口右键单击“按左键”事件“动作”列上的白色闪电，在快捷菜单中选择“VBS 动作…”，打开 VBS 动作编辑器。在代码区输入如下程序：

```
Sub OnLButtonDown (ByVal Item, ByVal Flags, ByVal x, ByVal y)
    Dim Recht
    Dim cmdButton
    Set Recht  =  ScreenItems("静态文本 1")
    Set cmdButton = ScreenItems("按钮 2")
    If Recht. Visible = 0 Then         判断静态文本的显示状态
```

```
    Recht. Visible = 1 显示帮助信息
    cmdButton. Text = "隐藏帮助"   修改按钮显示文本
  Else
    Recht. Visible = 0   隐藏帮助信息
    cmdButton. Text = "显示帮助"   修改按钮显示文本
  End If
End Sub
```

4）单击“语法检查”按钮，如有错误提示，修改错误后重新进行检查，直至没有错误提示，单击“确定”按钮，完成 VBS 动作的编辑，退出 VBS 动作编辑器。

保存画面，在 WinCC 图形编辑器或项目管理器中激活项目，测试组态的结果。

（4）采集变量的状态

```
Dim myTag, last_err
Set myTag  =  HMIRuntime. Tags("Istwert_1")
last_err  =  myTag. LastError    读取变量 Istwert_1 的状态
If (last_err  >  0) Then
  Msgbox myTag. QualityCode   在消息框中显示变量的质量代码
End If
```

（5）读取 WinCC 变量的值

```
Dim myTag, value1
Set myTag  =  HMIRuntime. Tags("Istwert")
myTag. Read    'Read 方法读取变量 Istwert 并将变量值存放到 Value 属性中
value1 =  myTag. Value   对变量的值进行处理
```

（6）写一个值到 WinCC 变量

```
Dim my_Soll1
Set my_Soll1 = HMIRuntime. Tags ("Soll1")
my_Soll1. Value  =  0      通过变量的 Value 属性给变量赋值
my_Soll1. Write (vbFalse)    'Write 方法写变量值到变量
```

注意：在变量的读取或写入的过程中，用到一些属性和方法，如 Read 和 Write 方法以及变量的 Value 属性等。一些参数可以改变读取或写入的方式，例如：“myTag. Read, 1”中的参数“1”表示直接从自动化系统中读取，而“my_Soll1. Write (vbFalse)”中的参数“vbFalse”表示直接将变量值写入自动化系统。

## 4.5　画面模板

### 4.5.1　画面模板的概念

画面模板是 WinCC 组态过程画面时的常用工具。对于实际项目中多个相同参数的设备，例如电机、驱动器和开环控制器等，使用画面模板，可以避免反复组态具有相同参



数的设备，降低组态和管理类似设备的成本。画面模板与结构类型变量配合使用，可以实现在一个过程画面中根据条件显示和控制具有相同类型参数的多个设备，极大地丰富过程画面的信息。结构类型变量的相关概念可以参考第 3 章 3.2.2 小节“5. 结构类型变量”和 3.2.4 小节“3. 结构类型变量的创建”。

画面模板本质上就是一个 WinCC 过程画面，只是画面模板在 WinCC 运行系统中必须借助 WinCC 的“画面窗口”（Picture Window）对象（该对象在图形编辑器的对象选项板的“智能对象”列表中）在其他的过程画面中显示，此过程称为“画面窗口”（Picture Window）对画面模板的调用。与高级语言中函数或子程序的调用过程类似，如果对画面模板的每一次调用赋予不同的参数值，那么“画面窗口”将显示不同的结果。根据这个原理，如果参照一个结构类型组态一个画面模板（例如参照第 3 章 3.2.4 小节“图 3－21”和“图 3－22”创建的结构类型 motor 组态一个电机画面模板），且以该结构类型的变量为调用时必须提供的参数，那么对该画面模板的每一次调用将在“画面窗口”中显示不同的结构类型变量的值（例如参数为 motor1 时将在画面窗口中显示 motor1 的所有变量值）。

与结构类型变量配合，WinCC 可以通过变量前缀（Tag Prefix）、间接寻址和动态向导等方式实现“画面窗口”对画面模板的每次调用都只显示和处理对应的结构变量的数据。本节将以第 3 章 3.2.4 小节“图 3－21”和“图 3－22”创建的结构类型 motor 为例，介绍变量前缀方式的组态过程。

### 4.5.2　变量前缀的概念

由第 3 章 3.2.4 小节“图 3－24”可知，创建名称为“motor1”的结构类型 motor 的实例时，会自动生成 4 个结构类型变量，即“motor1. set”、“motor1. actual”、“motor1. start”和“motor1. auto”。这些结构变量主要由三个部分组成，即：结构类型实例名＋“.”＋结构类型的元素名，例如“motor1. set”由结构类型实例名“motor1”、“.”和结构类型元素名“set”组成。对于一个结构类型变量，其变量前缀为结构类型实例名＋“.”，由此概念可知，结构类型变量“motor1. set”、“motor1. actual”、“motor1. start”和“motor1. auto”的变量前缀均为“motor1.”。

### 4.5.3　变量前缀方式应用画面模板的原理

前面已经提到，结构类型变量和画面模板的结合，可以实现在一个过程画面中根据条件显示和控制具有相同类型参数的多个设备，具体的表现是“画面窗口”对象对画面模板的每一次调用，都只是显示和处理对应设备的数据。变量前缀方式实现该功能的原理包含三个方面：

① 组态与结构类型的元素名相同的内部变量。例如，针对结构类型 motor，组态 4 个变量名分别为“set”、“actual”、“start”和“auto”的内部变量。

② 组态画面模板的动态化时，使用与结构类型的元素名相同的内部变量进行连接，连接包括“变量连接”、“直接连接”和“动态对话框”等动态化方法。例如，针对结构类型 motor 组态一个电机控制的画面模板，画面模板中组态一个“输入/输出域”用于设定电机的转速，则该“输入/输出域”应该与变量“set”进行连接。

③“画面窗口”对象调用画面模板时，应该静态或动态地修改其“画面名称”属性

和“变量前缀”属性。其中“画面名称”属性指定“画面窗口”对象所调用的画面模板，“变量前缀”属性用于确定“画面窗口”当前显示和控制的设备。“画面窗口”的“画面名称”和“变量前缀”属性如图 4-57 所示。

“画面窗口”调用画面模板时，WinCC 系统将“画面窗口”的“变量前缀”属性与所调用的画面模板中的画面对象所连接的变量结合，生成完整的结构类型变量，那么对应的画面对象将在“画面窗口”中显示和处理该结构类型变量的数据。例如，调用时“画面窗口”的“变量前缀”属性为“motor1.”，则与变量“set”连接的“输入/输出域”显示和处理结构类型变量“motor1. set”的数据，由此也可以简单地得出当前整个“画面窗口”显示和处理的是名称为“motor1”的设备的所有数据。




图 4-57　“画面窗口”对象的属性

## 4.5.4　变量前缀方式组态实例

下面以电机控制为例，说明变量前缀方式应用画面模板的具体方法和步骤。

**1. 创建结构数据类型和结构类型变量**

在工业自动化控制领域，类似于电机、驱动器和开环控制器等控制对象，往往需要对其包含的多个参数进行监控，创建结构数据类型和对应的结构变量是监控这类对象的最好选择。结构类型变量的相关概念请参考第 3 章 3.2.2 小节“5. 结构类型变量”和 3.2.4 小节“3. 结构类型变量的创建”，在此不再赘述。本节的电机控制实例将以第 3 章 3.2.4 小节“图 3-21”和“图 3-22”创建的结构类型 motor 为基础。

**2. 创建与结构数据类型的元素对应的内部变量**

由 4.5.3 可知，与结构数据类型的元素名相同的内部变量用于组态画面模板动态化。结构数据类型 motor 共包含 4 个元素，元素名分别为“set”、“actual”、“start”和“au-



to”，参考第 3 章 3.2.4 小节介绍的创建内部变量的方法，在 WinCC 资源管理器的“内部变量”文件夹下创建变量组和 4 个内部变量，如图 4-58 所示。

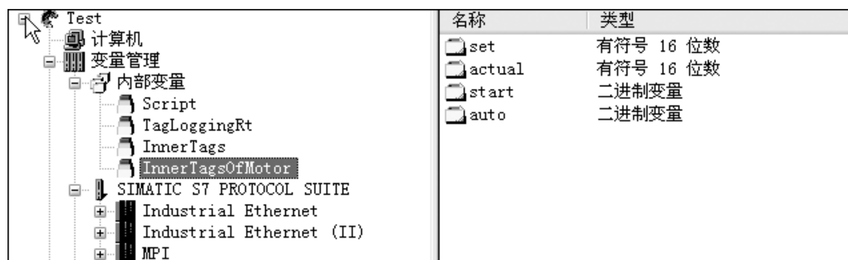



图 4-58　创建与数据类型 motor 的元素对应的内部变量

### 3. 组态电机控制的画面模板

如图 4-59 所示，电机控制的画面模板中包含了操作和显示电机的一些典型要素。标签“电机设定转速”下的“棒图”和“输入/. 输出域”用于显示电机在自动模式下的设定转速，其中通过“输入/输出域”可以由现场操作人员设定电机的转速；标签“电机实际转速”下的“棒图”和“输入/. 输出域”用于显示实时采集的电机实际转速；左边的选择开关用于启动和停止电机的运行；右边的选择开关用于设置电机的自动/手动运行模式。下面详细介绍该画面模板的组态过程。

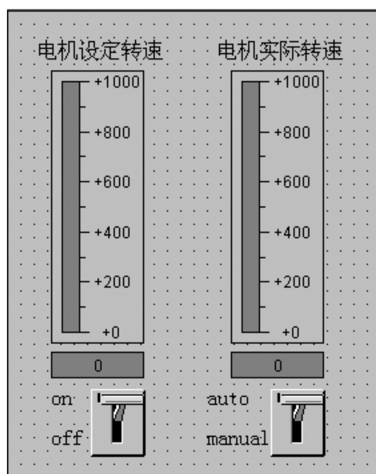



图 4-59　电机控制的画面模板

1）在 WinCC 的“图形编辑器”中新建一个画面，画面名称为“MotorPanel”，用鼠标右键单击该画面，在弹出的菜单中选择“属性”，打开“对象属性”对话框，并在“属性”选项卡中选择“几何”属性，将画面的“画面宽度”设置为 280，“画面高度”设置为 350，如图 4-60 所示。




图 4-60　新建画面“MotorPanel”

2）在 WinCC 的“图形编辑器”→“对象选项板”→“标准对象”中，选择“静态文本”添加到画面内，并将“文本”属性设置为“电机设定转速”；在“对象选项板”→“智能对象”中，选择“棒图”添加到画面内，并在弹出的“组态对话框”中，将该棒图与内部变量“set”建立连接；在“对象选项板”→“智能对象”中，选择“输入/输出域”添加到画面内，并在弹出的“组态对话框”中，将其图与内部变量“set”建立连接；打开“图形编辑器”的库，在“全局库”→“Operation”→“Toggle Buttons”中，将选择开关“On_Off_4”添加到画面中，用鼠标右键单击该选择开关，打开“对象属性”窗口，在“属性”选项卡中选择“UserDefined1”→“Toggle”属性，用鼠标右键单击“Toggle”属性“动态”列的白色灯泡，在弹出的菜单中选择“变量”，并将“Toggle”属性的动态值与变量“start”建立连接；在选择开关旁边添加两个“静态文本”，并将“文本”属性分别修改为“On”和“Off”。将以上添加的画面对象按照图 4-59 左半部分所示进行布局。

3）图 4-59 右半部分的组态过程与步骤 2）相同，可以按照步骤 2）中介绍的方法逐个添加画面对象并组态其动态化，更好的方法是将步骤 2）中组态的画面对象进行复制粘贴操作并对其属性进行针对性的修改。右半部分的“棒图”和“输入/输出域”连接的变量为“actual”，选择开关“On_Off_4”连接的变量为“auto”。

4）单击“图形编辑器”工具栏的“保存”按钮或“文件”菜单的“保存”命令，保存以上组态。

**4. 组态电机控制画面模板的调用**

前文已述及，画面模板虽然本质上就是 WinCC 画面，但是不会通过画面切换等方式单独地在 WinCC 运行系统中显示，而是在需要的情况下在其他画面中由“画面窗口”对象进行调用后显示，不需要时可以进行隐藏，即通常所说的“画中画”。“画中画”可以有效地提高 WinCC 过程画面的利用率，改善画面的美观性和友好性。

画面模板是为一些具有相同参数的多个设备的监控而组态的，例如现场多台电机的监控。“画面窗口”对画面模板的一次调用，将只显示对应的一台设备的数据，WinCC 通过定义“画面窗口”的“变量前缀”属性与结构变量的变量前缀对应来实现画面模板的调用与设备的对应。“画面窗口”的“变量前缀”属性可以是静态定义和动态定义两种方式，静态定义方式每一台设备需要组态一个“画面窗口”对象，如图 4-61 所示，而动态定义方式可以多台设备共用一个“画面窗口”对象，如图 4-62 所示。显然动态定义方式画面利用率更高，但是组态较静态定义方式复杂。下面分别介绍两种方式的组态方法。

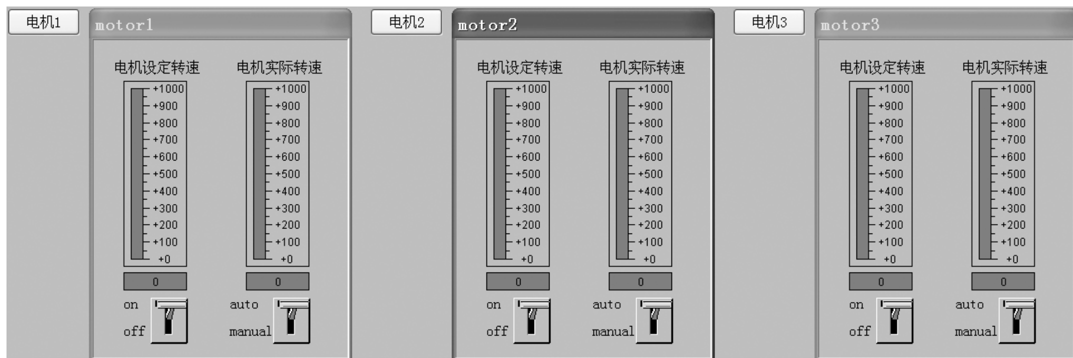


图 4-61　“变量前缀”属性静态定义方式画面模板的调用



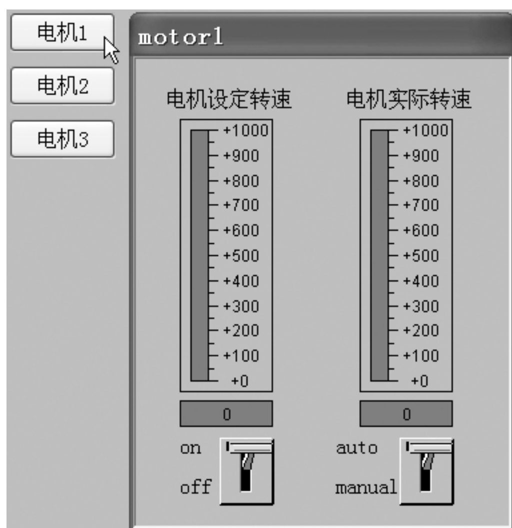



图 4-62 “变量前缀”属性动态定义方式画面模板的调用

（1）“变量前缀”属性静态定义方式

图 4-61 中，采用静态方式定义“画面窗口”的“变量前缀”属性，每台电机需要组态一个“画面窗口”对象，按钮用于切换“画面窗口”的显示和隐藏状态。详细组态过程如下：

1）参考第 3 章 3.2.4 小节图 3-23 展示的方法创建结构数据类型 motor 的实例（每个实例包含多个结构变量），实例数根据现场的电机数确定，一个实例对应 1 台电机，这里假设有三台电机，因此创建的实例为 motor1、motor2 和 motor3，如图 4-63 所示。

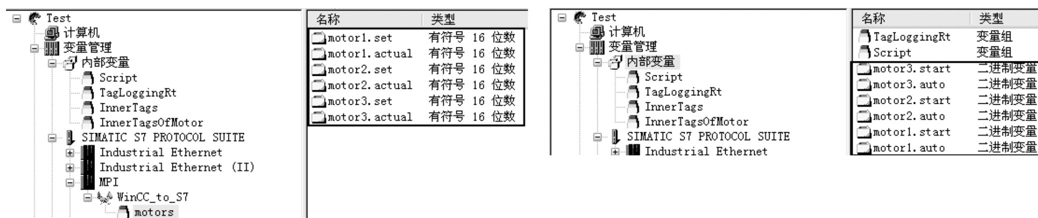



图 4-63 创建结构数据类型的实例

2）在 WinCC“图形编辑器”中新建一个画面，画面名称为“Motors”。在“图形编辑器”的“对象选项板”→“智能对象”中，选择“画面窗口”添加到画面内。用鼠标右键单击该画面窗口，在弹出的菜单中选择“属性”，打开“对象属性”窗口，按照图 4-65 设置“画面窗口”的属性。这里“窗口对象”的“画面名称”属性、“变量前缀”属性和“标题”属性均设置了相关的静态值。

3）在“图形编辑器”的“对象选项板”→“窗口对象”中，选择“按钮”添加到画面内，并将该按钮的文本设置为“电机 1”。用鼠标右键单击该按钮，在弹出的菜单中选择“属性”，打开“对象属性”窗口。激活属性窗口的“事件”选项卡，在左侧窗口选择“鼠标”，然后用鼠标右键单击右侧窗口“按左键”的“动作”列的白色闪电，在弹出的菜单中选择“C 动作...”，打开“编辑动作”窗口，如图所示 4-65 所示，在窗

口的代码区输入以下程序代码：

```
{
  BOOL a;
  a = GetVisible(lpszPictureName,"画面窗口 1");   //获取“画面窗口 1”的 显示 1/隐藏 0 状态
  a = ! a;                                        // 状态值取反
  SetVisible(lpszPictureName,"画面窗口 1",a);     //重新设置“画面窗口 1”的状态
}
```




图 4-64　设置“画面窗口”对象的属性

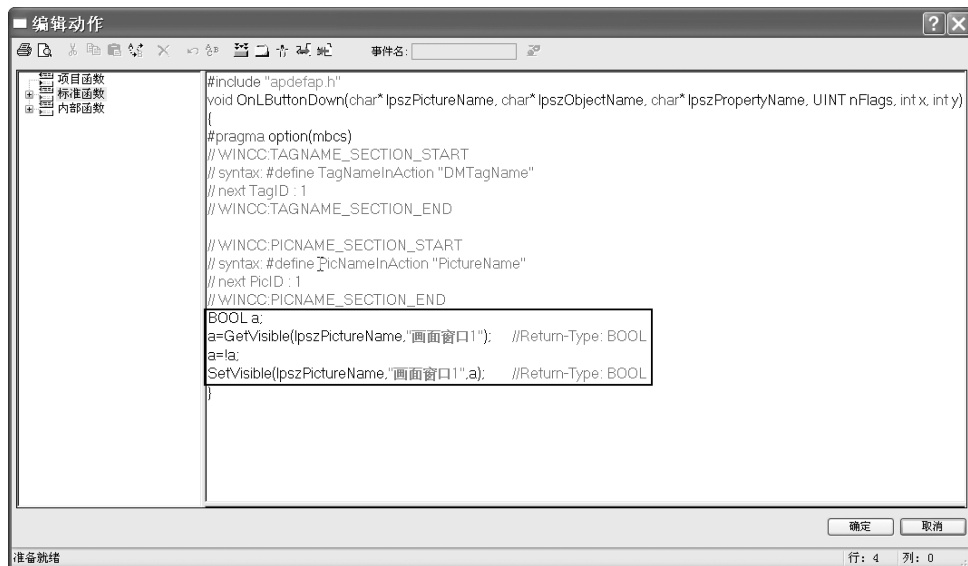



图 4-65　组态按钮的事件动作



4）按照步骤 3）介绍的方法组态另外两台电机的“按钮“和“画面窗口”，将一些相关位置的信息做针对性的修改（如“画面窗口 2”的“变量前缀”属性为“motor2.”，“标题”属性为“motor2”等），并按图 4-62 所示进行布局。

5）单击“图形编辑器”工具栏的“保存”按钮或“文件”菜单的“保存”命令，保存以上组态。

在 WinCC 图形编辑器或项目管理器中激活项目，测试上面的组态结果。测试过程中，改变某个实例（例如 motor1）的结构变量的值，改变结果将在该实例对应的“画面窗口”中（例如“画面窗口 1”）得以体现，而其他实例对应的“画面窗口”将没有任何变化。

（2）“变量前缀”属性动态定义方式

图 4-63 中，采用动态方式定义“画面窗口”的“变量前缀”属性，多台电机可以共享一个“画面窗口”对象，单击任一按钮将改变“画面窗口”当前所监控的电机，“画面窗口”的标题将作相应的改变。详细组态过程如下：

1）创建结构数据类型 motor 的实例，创建过程与“变量前缀”属性静态定义方式的组态过程步骤 1）相同，创建结果请参考图 4-64。

2）在 WinCC“图形编辑器”中新建一个画面，画面名称为“Motors”。在“图形编辑器”的“对象选项板”→“智能对象”中，选择“画面窗口”添加到画面内。用鼠标右键单击该画面窗口，在弹出的菜单中选择“属性”，打开“对象属性”窗口，按照图 4-66 设置“画面窗口”的属性。这里“窗口对象”的“画面名称”属性、“变量前缀”属性和“标题”属性均未设置静态值。




图 4-66 设置“画面窗口”对象的属性

3）在“图形编辑器”的“对象选项板”→“窗口对象”中，选择“按钮”添加到画面内，并将该按钮的文本设置为“电机 1”。用鼠标右键单击该按钮，在弹出的菜单中选择“属性”，打开“对象属性”窗口。激活属性窗口的“事件”选项卡，在左侧窗口

选择“鼠标”，然后用鼠标右键单击右侧窗口“按左键”的“动作”列的白色闪电，在弹出的菜单中选择“C 动作 ...”，打开“编辑动作”窗口，如图所示 4-67 所示，在窗口的代码区输入以下程序代码：

```
{
  SetTagPrefix(lpszPictureName,"画面窗口 1","motor1.");                  // 设置“变量前缀”属性
  SetPictureName(lpszPictureName,"画面窗口 1","MotorPanel.Pdl");  //设置“画面名称”属性
  SetPropChar(lpszPictureName,"画面窗口 1","CaptionText","motor1");// 设置“标题”属性
  SetVisible(lpszPictureName,"画面窗口 1",1);                            // 设置“显示”属性
}
```

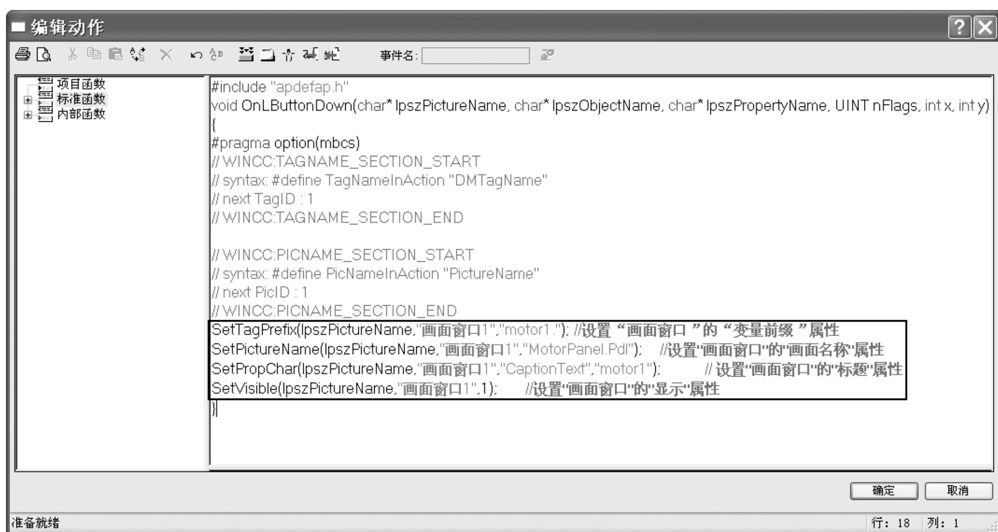



图 4-67　组态按钮的事件动作

4）按照步骤 3）介绍的方法组态另外两个按钮，将一些相关位置的信息做针对性的修改（如“变量前缀”属性分别为“motor2.”和“motor3.”，“标题”属性分别为“motor2”和“motor3”等），并按图 4-63 进行布局。

5）单击“图形编辑器”工具栏的“保存”按钮或“文件”菜单的“保存”命令，保存以上组态。

在 WinCC 图形编辑器或项目管理器中激活项目，测试上面的组态结果。

注意：无论是静态方式还是动态方式定义“画面窗口”的“变量前缀”属性，“变量前缀”属性的值必须与结构变量的前缀完全一致。

## 4.6　WinCC 用户权限管理

HMI 系统运行时，可能需要创建或修改一些重要的参数，例如修改电机转速的设定值、修改 PID 控制器的参数等。很显然，此类操作只允许指定的人员进行，禁止未经授权的人员访问系统的重要数据、修改系统的重要参数以及进行影响 WinCC 系统正常运行的操作，以保障生产的安全。



用户权限管理主要是组态不同层次的用户来管理生产过程，对于不同的用户，系统管理员可以根据需要授予用户各自的权限并设置相应的密码，系统因此有了分层的访问保护。

WinCC 用户管理器的主要功能就是分配用户，指派和管理用户的访问权限，以便杜绝未经授权的访问。这样，每个过程操作、档案操作以及 WinCC 系统操作都可对未经授权的访问加锁。一个用户最多可以分配 999 种权限，也可以在运行系统中进行用户权限的分配。

## 4.6.1 WinCC 用户管理器

WinCC 用户权限管理分为两个部分：用户权限管理的组态系统和用户权限管理的运行系统。用户权限管理组态系统即 WinCC 用户管理器，用来创建和管理用户、分配用户权限和设置用户密码等。当新用户被创建、分配权限和设置密码后，该用户的权限将记录在一个表格中。用户权限管理的运行系统主要任务是监控系统的登录和访问权限。

WinCC 用户管理器的主要任务概括如下：

① 创建、编辑用户（最多创建 128 个）和用户组（最多创建 10 个）。

② 分配和管理用户的访问权限。

③ 设置访问保护。

④ 有选择地防止未授权访问单个系统功能。

⑤ 指定时间长度用户未进行任何操作时使用户自动退出登录，防止未授权的访问。

在 WinCC 项目管理器的浏览窗口用鼠标右键单击“用户管理器”，在弹出的菜单中选择“打开”，将进入 WinCC 用户管理器，如图 4-68 所示。除了菜单栏、工具栏和状态栏，用户管理器主要包含浏览窗口和表格窗口两个窗口。通过浏览窗口可以查看已经建立的用户组和用户。表格窗口的上半部分显示了在浏览窗口选择的用户以及与用户有关的所有参数设置，下半部分显示了为当前选择的用户设置的权限。

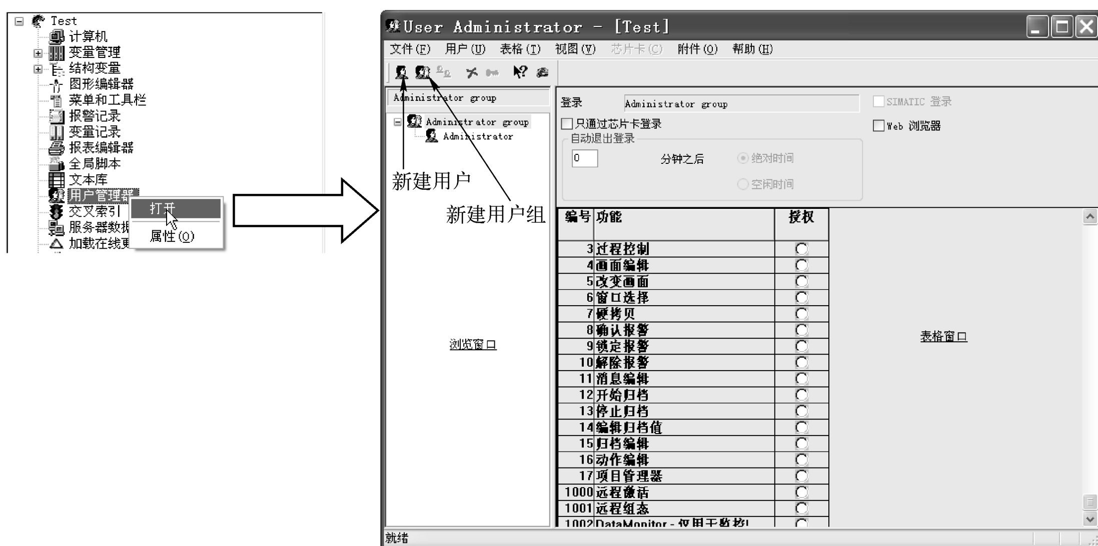



图 4-68　WinCC 用户管理器

在与用户有关的参数设置中，“自动退出登录”项用于设置用户登录运行系统后的退出方式，避免用户登录后未退出被其他人员通过此用户访问运行系统。如果时间框中输入为“0 分钟之后”，则用户直到系统关闭访问后才退出登录。组态的自动退出登录时间从用户登录时开始计算，分为绝对时间和空闲时间，绝对时间表示达到设定的时间后用户将无条件的自动退出登录，空闲时间表示用户在该时间长度内没有任何操作时自动退出登录。如果只允许用户通过芯片卡登录，则需要勾选“只通过芯片卡登录”复选框。

首次打开“用户管理器”时，表格窗口中包含一些 WinCC 预置的授权。这些预置的授权包含标准授权和系统授权两部分，如图 4-69 所示，编号为 1 ~ 17 的是预置的标准授权，编号 1000 ~ 1002 的是 3 个预置的系统授权。

WinCC 预置的标准授权的含义见表 4-5。除了“用户管理”授权以外，其他标准授权都可进行删除或编辑。具有较低编号的授权不包含在具有较高编号的授权中，但将独立代替每个授权功能。“管理员组”的成员始终具有“用户管理”授权。标准授权将在组态系统中进行分配，但只在运行系统中产生作用。这将避免已登录的用户在运行系统中对所有系统区的未经许可的访问。

表 4-5　WinCC 预置的标准授权

| 编　号 | 授权名称 | 含　义 |
|---|---|---|
| 1 | 用户管理 | 该授权准许用户访问用户管理器。如果设置了该授权，则用户可以调用用户管理器并做改变 |
| 2 | 数值输入 | 该授权允许用户在 I/O 域中手动输入数值 |
| 3 | 过程控制 | 该授权将使用户能够完成控制操作，例如，手动/自动切换 |
| 4 | 画面编辑 | 该授权允许用户改变画面和画面元素（例如通过 ODK） |
| 5 | 切换画面 | 该授权允许用户触发画面切换，从而打开另一个已组态的画面 |
| 6 | 窗口选择 | 该授权允许用户在 Windows 中切换应用程序窗口 |
| 7 | 硬拷贝 | 该授权允许用户硬拷贝当前的过程画面 |
| 8 | 确认报警 | 该授权将使用户能够确认消息 |
| 9 | 锁定报警 | 该授权允许用户锁定消息 |
| 10 | 释放报警 | 该授权允许用户解锁（释放）消息 |
| 11 | 消息编辑 | 该授权允许用户改变报警记录中的消息（例如通过 ODK） |
| 12 | 启动归档 | 该授权允许用户启动归档过程 |
| 13 | 停止归档 | 该授权允许用户停止归档过程 |
| 14 | 编辑归档值 | 该授权允许用户组态归档变量的计算过程 |
| 15 | 归档编辑 | 该授权允许用户控制或改变归档过程 |
| 16 | 动作编辑 | 该授权允许用户执行和改变脚本（例如通过 ODK） |
| 17 | 项目管理器 | 该授权将允许用户访问 WinCC 项目管理器 |

1000 ~ 1099 之间的授权是系统授权。WinCC 预置的 3 个系统授权的含义见表 4-6，它们由系统自动生成，用户不能对其进行创建、修改或删除。正如任何其他授权一样，可将系统授权分配给用户。系统授权在组态系统中以及在运行系统中都是有效的。在组态系统中，系统授权将避免未在项目中登录的用户未经许可访问诸如服务器项目等操作。



表 4-6　WinCC 预置的系统授权

| 编　　号 | 授权名称 | 含　　义 |
|---|---|---|
| 1000 | 远程激活 | 如果存在该设置，则用户可从其他计算机上启动或停止该项目的运行系统 |
| 1001 | 远程组态 | 如果设置了该条目，则用户可从其他计算机上进行组态，并对项目进行修改 |
| 1002 | DataMonitor－仅用于监控！ | 如果设置了该条目，则用户从其他计算机上只能打开项目，而不能进行修改或执行控制操作 |

## 4.6.2　组态用户管理

组态用户管理系统时必须采取下列基本操作：

① 添加所需要的用户组。

② 设置用户组的相应授权。

③ 添加用户，并分配各自的登录名称和口令。当添加新的用户时，可对组的属性进行复制。此时，建议为用户分配具备相应授权（希望这些用户拥有的授权）的组。

④ 为各种不同用户选择特定的授权。此时可以设置用户自动退出登录的时间，该时间段结束以后，系统将自动注销用户，以防止发生未经授权的输入。也可确定用户是否能够只通过芯片卡进行登录，以及如果用户通过 Web 连接到系统时应使用哪些设置。

**1. 创建用户组和用户**

打开 WinCC 用户管理器后，可以在原有的管理员组（Administrator group）下建立新的管理员用户并设置密码。用鼠标右键单击浏览窗口的“管理员组”，在弹出的菜单中选择“添加用户”或选中“管理员组”后单击工具栏按钮，将弹出“添加新用户”对话框，如图 4-69 所示。在“登录”项后的文本框中输入新用户的用户名在（用户名要求不少于 4 个字符），在“密码”项后的文本框中输入新用户的登录密码（密码要求不少于 6 个字符），在“验证密码”项后的文本框中输入的密码必须与“密码”项设置的密码完全一致，否则将出现错误提示。单击“确定”按钮完成新用户的创建，新创建的用户在浏览窗口的管理员组下可以查看。




图 4-69　添加新用户对话框

如果创建的用户比较多，可以建立不同的用户组对用户进行管理。对于一个成熟的用户管理系统，只能有少数几个用户属于管理员组。在用户管理器的浏览窗口用鼠标右键单击在弹出的菜单中选择“添加组”或单击工具栏的“”按钮，将自动添加一个用户组到浏览窗口中，可以对用户组的名称进行修改。添加新的用户组后，可以在组中按照上面介绍的方法添加用户，也可以为新建的用户组进行授权，在组中创建的用户可继承用户组所设置的授权，也可单独进行授权。

2. 为用户组和用户设置授权

创建新的用户组和新的用户后，需要为其设置相应的授权。用户可以直接继承用户组所设置的授权，也可以单独进行授权。在用户管理器的浏览窗口选中需要设置授权的用户组或用户，双击表格窗口中要设置给当前用户组或用户的授权后的圆圈，当小圆圈变为红色时，该授权被选中，表示此用户组或用户拥有该项授权，如图 4-70 所示。名称为“Operator”的用户组具有“改变画面”和“窗口选择”两项授权。

用户拥有不同的授权，其权限级别是不同的，权限的高低主要从两个方面体现：一个方面是拥有的授权数量，另一方面是拥有的授权的重要程度。如果当前用户拥有而其他用户没有这个授权，该用户的权限级别自然就高。

除了 WinCC 预置的标准授权和系统授权，还可以根据项目的需要添加自定义的授权。单击用户管理器的菜单表格→“插入授权”，将打开“插入行号”对话框，可以设置新添加的授权的行号，单击“确定”按钮后，将在表格窗口的指定行号增加一项授权，接下来可以编辑该项授权的功能。

除了“用户管理”授权和系统授权，用户选中表格窗口中任一项授权，通过单击用户管理器的菜单表格→“删除授权”，可以删除该项授权。




图 4-70　设置用户组或用户的授权

### 4.6.3　组态画面对象的访问保护

在用户管理器中组态用户组和用户的授权后，接下来就要在过程画面中对那些需要设置访问保护的画面对象进行组态。这里以本章 4.4.2 节图 4-21 组态的“退出 WinCC 运行系统”按钮设置访问保护，使只有拥有“过程控制”授权的用户单击该按钮才起作用



为例，详细介绍组态画面对象的访问权限的方法。

用鼠标右键单击“退出 WinCC 运行系统”按钮，在弹出的菜单中选择“属性”，打开“对象属性”窗口，如图 4-71 所示。激活“属性”选项卡，在左侧窗口选择“其他”，右侧窗口选择“授权”属性项。可以看到，默认情况下，“授权”属性的静态值是“<没有授权保护>”。用鼠标双击或右键单击“<没有授权保护>”，在弹出的菜单中选择“编辑”，打开“授权”对话框，在授权列表中选择“过程控制”授权，如图 4-71 所示，单击“确定”按钮退出对话框，该按钮的“授权”属性的静态值将更新为“过程控制”，完成了访问保护的组态。

当 WinCC 项目运行时，只有拥有“过程控制”权限的用户登录后才能操作“退出 WinCC 运行系统”按钮，没有“过程控制”权限的用户登录后单击此按钮，将提示“没有许可权”。

过程画面中其他画面对象的访问保护组态与上述过程类似。




图 4-71　设置按钮的访问权限

## 4.6.4　组态用户登录和注销的对话框

WinCC 项目运行后需要弹出用户登录的对话框，以用户管理器中组态的用户名和密码登录后，才可以操作设置了访问保护的画面对象。对于一个成熟的 WinCC 项目来说，没有用户登录时，或执行注销操作运行系统恢复到没有用户登录的状态时，将只能观看过程画面，不能进行任何操作。

弹出用户登录对话框可以通过两种方式实现，一种方式是通过按下预先设置热键来调出登录对话框，另一种方式通过单击按钮弹出登录对话框，该按钮的鼠标“单击”事

件必须组态了相应的脚本程序。

### 1. 使用热键

在 WinCC 项目管理器的浏览窗口用鼠标右键单击项目名称，在弹出的菜单中选择“属性”，打开“项目属性”对话框，激活“热键”选项卡，可以设置用户登录和注销的热键，如图 4-72 所示。

选中“动作”列表中的“登录”，鼠标单击“分配”按钮下的文本框，同时按下要分配给“登录”动作的热键，例如〈Ctrl〉和〈L〉键，这时两个键将出现在文本框中，如图 4-73 所示，单击“分配”按钮即可以将该热键分配给“登录”动作。如果动作没有设置热键，选中该动作时文本框中将显示“无”。

用同样的方法可以给“退出”（用户注销）动作分配热键，例如〈Ctrl + E〉的组合。则项目运行时可以通过按下〈Ctrl + F〉来弹出用户登录对话框进行登录，按下〈Ctrl + E〉进行用户注销。

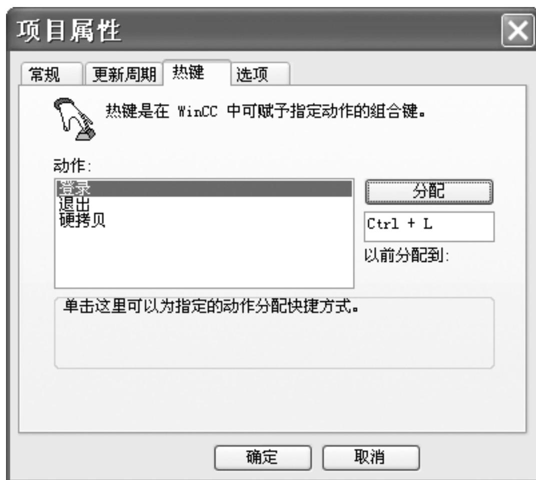



图 4-72　设置用户登录和注销的热键

### 2. 使用按钮

WinCC 项目运行后，也可以通过单击按钮来弹出用户登录对话框，但必须为该按钮的鼠标“单击”事件组态相应的脚本程序。

在画面中添加两个按钮，其文本分别为“登录”和“注销”。为“登录”按钮组态的鼠标“单击”事件的 C 脚本程序如图 4-73 所示。为“注销”按钮组态的鼠标“单击”事件的 C 脚本程序如图 4-74 所示。




```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
#pragma code("UseAdmin.DLL")
#include "pwrt_api.h"
#pragma code()
PWRTLogin(1);
}
```

图 4-73　登录按钮的 C 脚本程序



注销

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
#pragma code("UseAdmin.DLL")
#include "pwrt_api.h"
#pragma code()
PWRTLogout();
}
```

图 4-74 注销按钮的 C 脚本程序

按钮的 C 动作组态方法请参考本章 4.4.6 小节“C 动作”的内容。

### 4.6.5 使用与用户登录相关的内部变量

在 WinCC 项目运行时，如果希望在过程画面或报表中显示已登录的用户的相关信息，可以使用系统提供的两个相关的内部变量，见表 4-7。

表 4-7 与用户登录相关的内部变量

| 变 量 名 称 | WinCC 用户管理器中的描述 |
|---|---|
| @ CurrentUser | 用户 ID |
| @ CurrentUserName | 用户名 |

在过程画面中插入一个 I/O 域，将 I/O 与@ CurrentUser 或@ CurrentUserName，建立变量连接，并将该 I/O 域的数据格式设置为字符串。项目运行时可以看到，当有用户登录时，当前登录的用户的用户名显示在此 I/O 域中。

## 4.7 组态技巧与技术问答

本节将介绍一些组态 WinCC 过程画面及其动态化的技巧，并针对过程画面及其动态化过程中遇到的一些常见的技术问题进行分析和解答。

### 4.7.1 应用 S7 – PLCSIM 和 WinCC 变量模拟器测试 WinCC 项目

在 WinCC 监控系统没有连接过程外围设备或连接了过程外围设备但过程没有运行的情况下，要对 WinCC 项目的组态结果和功能进行测试，可以应用西门子提供的 S7 – PLCSIM 仿真软件和 WinCC 的变量仿真工具“WinCC Tag Simulator”。

1. 应用 S7 – PLCSIM 测试 WinCC 项目

S7 – PLCSIM 仿真软件是西门子公司提供的可选软件包，在设计人员身边没有 PLC 等硬件时，用于模拟 PLC 硬件进行控制程序的测试。S7 – PLCSIM 可以直接模拟西门子 S7 – 300/400 系列 PLC 的运行，并测试程序。西门子 STEP 7 专业版包含了 S7 – PLCSIM 仿真软件，STEP 7 标准版需要在安装 STEP 7 后再安装 S7 – PLCSIM，S7 – PLCSIM 会自动嵌入 STEP 7 中。S7 – PLCSIM 仿真软件需要安装授权。

只有 WinCC 监控系统连接的过程外围设备是以西门子 S7 – 300/400 为主控制器时，才能采用 S7 – PLCSIM 测试 WinCC 项目，而且 S7 – PLCSIM 只能用来模拟 WinCC 项目中

创建的过程变量。

（1）打开 S7 – PLCSIM 仿真软件

可以通过两种方法打开 S7 – PLCSIM 仿真软件：一种是在 STEP 7 软件主界面“SIMATIC Manager”的工具栏中，单击“打开/关闭仿真器”按钮，打开 S7 – PLCSIM；另一种方法是在 Windows 的“开始”菜单中，选择“所有程序”→“SIMATIC”→“STEP 7”→“S7 – PLCSIM Simulating Modules”，打开 S7 – PLCSIM。

打开 S7 – PLCSIM 后，在其主界面中会自动出现 CPU 视图对象，单击工具栏中的输入、输出、位存储等按钮，在工作区内会生成相应的视图对象，如图 4–75 所示。在试图对象左上角的小窗口中可以设置视图对象的地址（例如 QB 6），单击视图对象右上角小窗口的选择按钮，可以设置按位（Bits）、二进制（Binary）、十进制（Decimal）、十六进制（Hex）和 BCD 码等格式输入和显示数据。

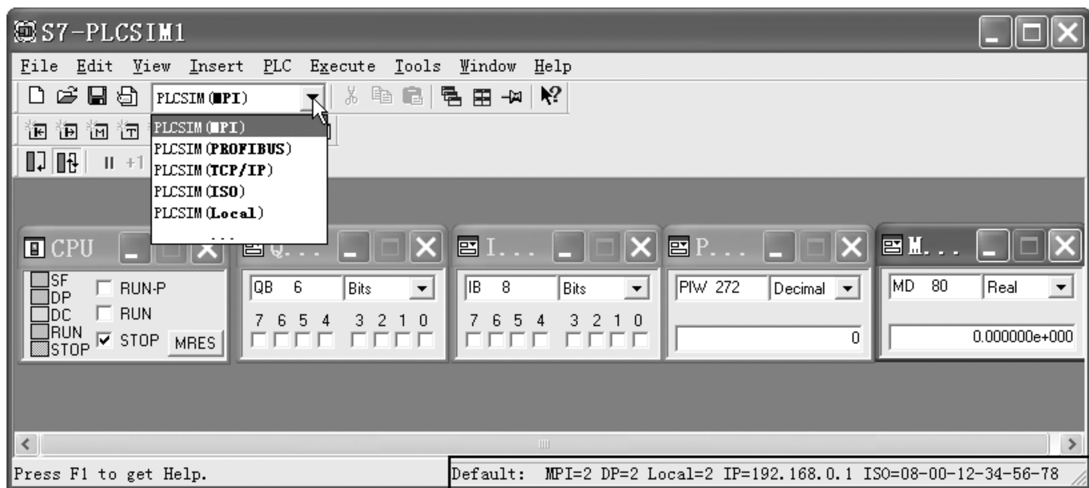


图 4–75 S7 – PLCSIM 主界面

（2）设置 S7 – PLCSIM 的通信接口

单击 S7 – PLCSIM 主界面的工具栏的下拉列表 PLCSIM(MPI)，可以设置 WinCC 项目与 S7 – PLCSIM 的通信接口，S7 – PLCSIM 为各种接口提供了默认的通信接口地址，可以在主界面右下角的状态栏中查看这些默认地址，如图 4–76 所示。

在测试 WinCC 项目时，WinCC 项目逻辑连接参数“S7 网络地址”的“站地址”一项（参考第 3 章 3.2.3 小节的图 3–14）必须与 S7 – PLCSIM 通信接口地址一致，才能保证 WinCC 与 S7 – PLCSIM 之间的通信连接正常。

（3）测试 WinCC 项目

测试 WinCC 项目时，先将 PLC 控制程序下载到 S7 – PLCSIM 中，然后激活 CPU 的运行状态（在 S7 – PLCSIM 主界面的 CPU 对象视图中勾选“RUN – P”或“RUN”）。接下来可以运行 WinCC 项目，测试 WinCC 与 S7 – PLCSIM 的通信状态，如果通信正常，就可以在 S7 – PLCSIM 主界面中通过改变输入变量的 ON/OFF 状态或数值的大小，控制 PLC 控制程序的运行过程，一方面可以监视 PLC 控制程序的运行结果，另一方可以同时在 WinCC 运行系统观察有关输出变量的状态（如指示灯的状态、“输入/输出域”数据的变



化等）来测试 WinCC 项目的组态结果和功能。

**2. 应用 WinCC 变量模拟器测试 WinCC 项目**

WinCC 变量模拟器（WinCC Tag Simulator）既可以用来模拟内部变量，又可以模拟过程变量。在没有连接过程外围设备时，只能模拟内部变量。如果已经连接了过程外围设备，过程变量的值将由变量模拟器直接提供，这样可以使用户用原有的硬件对 WinCC 项目进行功能测试。

注意：变量模拟器将过程值写入到所连接的自动化系统，这意味着必须考虑所连接过程外围设备的可能相应。

（1）模拟没有过程外围设备时的过程变量

在没有连接外围过程设备的情况下，只能模拟内部变量。为了对过程进行离线模拟，需要按下列步骤进行操作：

1）通过复制 WinCC 项目（复制方法参考第 3 章 3.3.2 小节的内容）的方法创建一个备份项目并将其重命名（例如“原项目名_sim”）。使用备份项目作为测试对象，并用该备份项目打开 WinCC 项目管理器，目的是保留原项目中的过程变量。

2）使用就“剪切”和“粘贴”功能将所要模拟的过程变量添加到内部变量中。切勿使用“复制”和“粘贴”，否则，WinCC 项目管理器为了确保变量名在项目中是唯一的，会自动生成变量名称的扩展名，这就与要模拟的变量名不相同了。过程变量添加到内部变量中，其地址信息将丢失。

这样可以在 WinCC 变量模拟器的帮助下为这些变量提供数值。测试阶段结束后，可以继续执行原项目。

（2）模拟已连接过程外围设备时的过程变量

对于已经连接了过程外围设备但过程没有运行的情况，过程变量的值可以由 WinCC 变量模拟器直接提供。这样可以使用户应用原有的硬件对 WinCC 项目进行功能测试，例如：

1）检查限制等级、消息输出。

2）测试报警、警告、出错消息的连续性，并检查状态显示。

3）预置、读取和修改数字量和模拟量的输入和输出。

4）报警模拟。

（3）使用 WinCC 变量模拟器测试 WinCC 项目

在 Windows 的“开始”菜单中，选择“所有程序”→“SIMATIC”→“WinCC”→“Tools”→“WinCC Tag Simulator”，打开 WinCC 变量模拟器，如图 4-77 所示。

注意：只有 WinCC 处于运行状态，变量模拟器才能起作用。

WinCC 变量仿真器主界面中包含“Lists of Tags”和“Properties”两个选项卡。选中“Lists of Tags”选项卡时，“Start Stimulation”按钮用于启动变量的仿真。“Tags”框中列出了要仿真的变量，如图 4-76 所示。选中“Properties”选项卡，可以设置要仿真的变量的属性，主要是设置变量的仿真函数以及与仿真函数相关的参数。WinCC 变量模拟器提供了 6 种仿真函数，如图 4-77 所示，包括：

- Sine（正弦）。
- Oscillation（振荡）。

- Random（随机）。
- Inc（加 1）。
- Dec（减 1）。
- Slider（滚动条）。

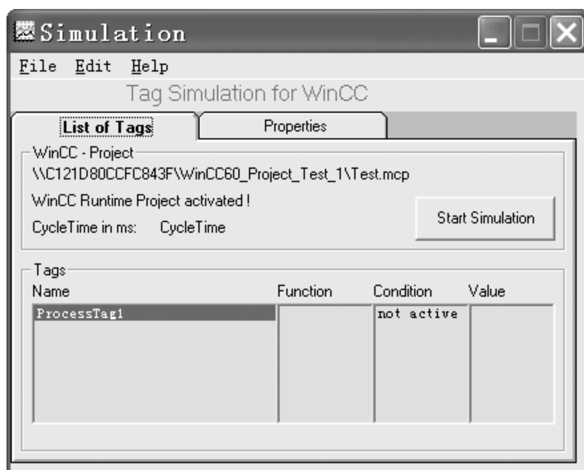



图 4-76　WinCC 变量仿真器

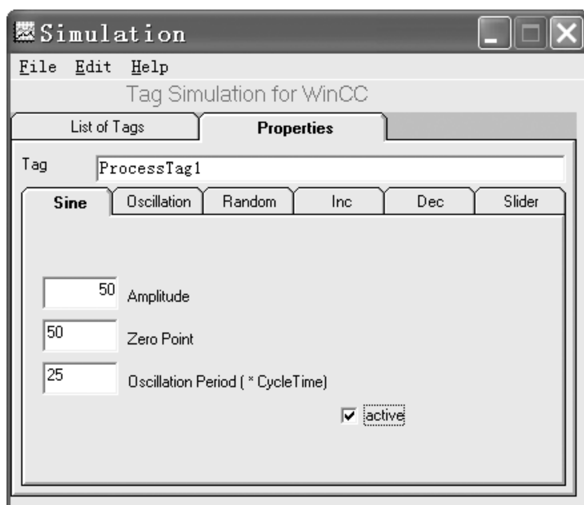



图 4-77　WinCC 变量模拟器的仿真函数

应用 WinCC 变量仿真器测试 WinCC 项目的步骤如下：

1）单击 WinCC 变量仿真器的“Edit”菜单的“New Tag”项，打开“变量”对话框，添加要仿真的 WinCC 变量。

2）选中“List of Tags”选项卡，在“Tags”框中选择一个已经添加的 WinCC 变量，然后选择“Properties”选项卡，设置该变量的仿真函数以及相关的参数，并勾选该仿真函数的“active”。

3）重复步骤 2），完成所有要仿真的变量的仿真函数和相关参数的设置。



4）在“List of Tags”选项卡中单击“Start Simulation”按钮，开始变量的仿真。

5）返回 WinCC 运行系统，打开相关的过程画面，监视变量仿真开始后的相关变化，测试 WinCC 项目的组态结果和功能。

### 4.7.2 WinCC 过程画面显示系统信息的组态方法

在 WinCC 的系统信息（System Info）通道的通信驱动程序下创建的 WinCC 变量，可以专门用于记录系统信息，因此在 WinCC 过程画面中创建画面对象（例如 I/O 域）与这样的 WinCC 变量建立连接，就可以在过程画面中显示系统信息。

系统信息通道的功能包括：在过程画面中显示时间；通过在脚本中判断系统信息触发事件；在趋势图中显示 CPU 负载；显示和监控客户机系统中不同服务器上可用的驱动器空间；监控可用磁盘容量并触发报警等。

系统信息通道可用的系统信息如下：

- 日期、时间：以 8 位字符集表示的文本型变量，可用各种不同的日期、时间格式表示。
- 年、月、日、星期、时、分、秒、毫秒：16 位无符号数变量，星期也可以用 8 位字符集的文本变量来表示。
- 计数器：32 位数，可设置起始值和终止值，该类型变量按从最小更新周期加 1 计数。
- 定时器：32 位数，可设置起始值和终止值，该类型变量按每秒加 1 计数。
- CPU 负载：32 位浮点数，可显示 CPU 负载时间或空闲时间的百分比。
- 空闲驱动器空间：32 位浮点数可表示本地硬盘或软盘的可用空间或可用空间百分比。
- 可用的内存：32 位浮点数，可表示空闲的内存量或内存量百分比。
- 打印机监控：无符号 32 位数，可显示打印机的一些状态信息。

系统信息通道无硬件要求，因为通道直接访问计算机上已安装的的系统信息。为了使通道起作用，必须和“SIMATIC S7 Protocol Suite. chn”通道一样建立通信连接。在系统信息通道的驱动程序下创建的 WinCC 变量虽然也是过程变量，但是不需要许可证，因此不算在授权计数中。

下面以在过程画面中应用 I/O 域显示系统的日期和时间为例，介绍在过程画面中显示系统信息的详细组态步骤：

1）用鼠标右键单击 WinCC 项目管理器浏览窗口的“变量管理”，在弹出的菜单中选择“添加新的驱动程序”，打开“添加新的驱动程序”对话框，如图 4-78 所示。选择“System Info. chn”，单击“打开”按钮，在“变量管理”中增加“系统信息”项。

2）用鼠标右键单击“变量管理”中的“系统信息”项，在弹出的菜单中选择“新驱动程序的连接”，打开“连接属性”对话框，如图 4-79 所示。在“名称”后的文本框中输入连接名称（例如“SystemInfo”），单击“确定”按钮，创建一个系统信息通道的连接“SystemInfo”。

3）用鼠标右键单击新建的连接“SystemInfo”，在弹出的菜单中选择“新建变量”，打开“变量属性”对话框，在“名称”框中输入变量名称（例如“SysTime”），单击“地址”

框后的“选择”按钮，打开“系统信息”对话框，设置该变量的格式，如图 4-80 所示，在“函数”后的下拉列表框中选择“日期”函数，在“格式化”后的列表中选择日期的格式，单击“确定”按钮返回到“变量属性”对话框，根据需要设置变量的其他属性后单击“确定”按钮，完成变量的创建并以同样的方法创建另一个系统信息变量“SysDate”。




图 4-78　添加系统信息通道的驱动程序




图 4-79　创建系统信息通道的连接并设置属性




图 4-80　创建系统信息变量并设置其格式



4）在过程画面中添加两个 I/O 域，将其“数据格式”属性修改为字符串，将两个 I/O 域分别与上面创建的系统信息变量“SysTime”和“SysDate”建立变量连接。保存并运行，就可以在该过程画面的两个 I/O 域中显示系统的日期和时间。

### 4.7.3　禁用 Windows 组合键的组态方法

对于投入正式运行的 WinCC 项目来说，为了保证 WinCC 项目的运行状态不被人为的不在计划范围内的中断或破坏，很多 Windows 的操作都是禁止的，因此在 WinCC 正式投入运行之前，要在 WinCC 组态系统中禁用 Windows 的组合键。禁用 Windows 组合键的方法如下：

1）在 WinCC 项目管理器的浏览窗口选中“计算机”，在数据窗口用鼠标右键单击要禁用 Windows 组合键的计算机，在弹出的菜单中选择“属性”，打开“计算机属性”窗口。

2）在“计算机属性”窗口激活“图形运行系统”选项卡，在“窗口属性”框中勾选“全屏”前的复选框，在“关闭”框中勾选“Alt + F4”、“最小化”、“最大化”和“关闭”前的复选框，如图 4-81 左图所示。

3）在“计算机属性”窗口激活“参数”选项卡，在“禁止键”框中勾选“Ctrl + Alt + Del”、“Alt + TAB”、“Ctrl + Esc”和“Alt + Esc”前的复选框，如图 4-81 右图所示。

4）单击“确定”按钮，关闭“计算机属性”对话框。




图 4-81　禁用 Windows 组合键

### 4.7.4　在 Windows 组合键被禁用时从 WinCC 运行系统进入组态系统

禁用 Windows 组合键的目的是为了保证 WinCC 项目的运行状态不被人为的不在计划范围内的中断或破坏，但在对 WinCC 系统进行维护时，工程技术人员需要从 WinCC 运行系统切换到 WinCC 组态系统（即进入 WinCC 项目管理器）。最好的方法是在过程画面中组态一个按钮，该按钮需要组态用户管理系统的访问权限保护（组态方法请参考本章 4.6 节的相关内容），并在该按钮的鼠标动作中组态下面的 C 脚本程序：

```
{
  #pragma code("user32.dll")
    BOOL SetForegroundWindow(HWND);
  #pragma code()
  HWND handle;
  handle = FindWindow("WinCCExplorerFrameWndClass", NULL);
  if (! SetForegroundWindow(handle)) printf("\r\n SetForeground fails");
}
```

### 4.7.5　二元开关操作的组态方法

二元开关操作是通过 WinCC 的画面对象（例如“窗口对象”中的“按钮”对象）执行开和关的操作，控制现场设备的动作（例如阀门的打开和关闭、电机的启动和停止）。对于现场来说，存在多种类型的开关，例如带确认操作的二元开关、不带确认操作的二元开关和行程开关等，WinCC 可以采用不同的动态化方法组态不同类型的开关，表 4-8“On/Off 操作基本原理”列出了现场的开关类型和对应的 WinCC 组态解决方案。

表 4-8　On/Off 操作基本原理

| 开关类型 | 功能描述 | 解决方案 | 备　注 |
|---|---|---|---|
| On/Off | 开关功能，使用鼠标单击，有确认 | C－动作 | |
| Ackn. | | 直接连接或动态向导或 C－动作 | |
| On/Off | 开关功能，使用鼠标单击，没有确认 | C－动作 | |
| On | 使用鼠标单击，在 var1 中置位，在 var2 中复位 | C－动作 | 行程开关，var1 和 var2 不能同时置位 |
| | 使用鼠标单击，在 var2 中置位，在 var1 中复位 | | |
| Off | 单击 On 按钮置位开关、单击 Off 按钮复位开关 | 直接连接或动态向导或 C－动作 | |
| On/Off | 鼠标左键单击用于开/右键单击用于关 | 直接连接或动态向导或 C－动作 | |

对 WinCC 来说，最好的解决方案是“直接连接”，因此，如果可以的话，应当使用“直接连接”方法进行组态。

对一个单元的开关操作应当直接在过程画面中进行，或者在一个两步操作（画面窗口）中进行。两步操作第一步是用一个按钮打开开关所在的操作面板（操作面板在画面窗口中显示），第二步就是在操作面板中操作开关，操作完成后可以关闭操作面板。

本节将详细介绍“有确认的二元开关”和“没有确认的二元开关”的 WinCC 组态方法和步骤，其他类型二元开关的组态方法可以以此为参考。

**1. 有确认的二元开关**

（1）任务定义

如图 4-82 所示，在过程画面中通过鼠标单击一个按钮（图中带黑色箭头的按钮）打开阀的操作面板（画面“VavlePanel1.PDL”在画面窗口中显示），操作面板包含一个按



钮（标识为“On/Off”的按钮）用来打开和关闭阀，但只有在单击一个确认按钮（标识为“Ack.”的按钮）后打开或关闭阀的动作才生效，单击标识“OK”的按钮可以关闭操作面板。执行打开阀的动作后，过程画面中的静态文本对象显示文本“阀打开”，同时圆背景闪烁；执行关闭阀的动作后，过程画面中的静态文本对象显示文本“阀关闭”，同时圆结束背景闪烁。

（2）组态步骤

1）组态二进制变量。在 WinCC 变量管理器中创建 2 个二进制类型的变量 ValveStatus 和 temp，ValveStatus 反映阀的当前状态（1：打开，0：关闭），temp 用做中间变量。

2）组态操作面板。在图形编辑器中新建一个画面，保存画面文件为“ValvePanel1. PDL”；添加 WinCC 对象选项板中的“标准对象”→“静态文本”，将属性“字体”→“文本”设置为“ValvePanel1”；添加 3 个 WinCC 对象选项板中的“窗口对象”→“按钮”，对象名分别为“按钮 1”、“按钮 2”和“按钮 3”，显示文本分别为“On/Off”、“Ack.”和“OK”；在“按钮 1”的事件“鼠标”→“按左键”处为“按钮 1”组态 1 个 C 动作，动作代码请参考下文；在“按钮 2”的事件“鼠标”→“按左键”处为“按钮 2”组态 1 个直接连接，将“来源”→“变量”→“temp”与“目标”→“变量”→“ValveStatus”建立连接；在“按钮 3”的事件“鼠标”→“按左键”处为“按钮 3”组态 1 个直接连接，将“来源”→“常数”→0 与“目标”→“当前窗口”→“显示”属性建立连接。

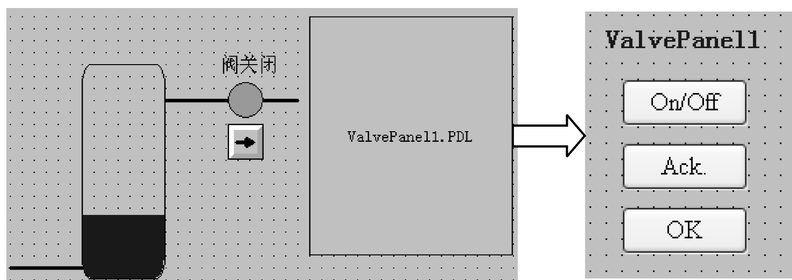



图 4-82 “有确认的二元开关”组态任务定义

3）组态过程画面中的对象。添加 1 个 WinCC 对象选项板中的“标准对象”→“圆角矩形”模拟液罐，将属性“填充”→“动态填充”设置为“是”，“填充量”属性设置为 30；添加 1 个 WinCC 对象选项板中的“标准对象”→“圆”，在属性“闪烁”→“闪烁背景”的“动态”列中组态一个与 ValveStatus 的变量连接，并将“更新周期”设置为“有变化时”；添加 1 个 WinCC 对象选项板中的“智能对象”→“画面窗口”到过程画面，对象名称为“画面窗口 1”，将属性“其他”→“显示”设置为“否”，“其他”→“边框”设置为“是”，“其他”→“调整大小”设置为“是”，双击“其他”→“画面名称”属性，将“画面名称”属性设置为“ValvePanel1. PDL”；添加 1 个 WinCC 库中的“全局库”→“Operation”→“Buttons 3D”→“Arrow Right”按钮，在事件“鼠标”→“鼠标动作”处为该按钮组态一个直接连接，将“来源”→“常数”→1 与“目标”→“画面中的对象”→“画面窗口 1”→“显示”属性建立连接；在圆上方添加 1 个 WinCC 对象选项板中的“标准对象”→“静态文本”，在属性“字体”→“文本”的动态列为

静态文本组态一个动态对话框，如图 4-83 所示，其中“事件名称”项设置为变量 ValveStatus 有变化时。




图 4-83　静态文本的动态对话框组态

4）保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

“按钮 1”的 C 动作代码如下：

```
{
  SetTagBit("temp",(short)! GetTagBit("ValveStatus"));
}
```

“按钮 1”的 C 动作代码的执行过程：取二进制变量 VavleStatus 的当前值，即阀的当前状态，由 GetTagBit（"ValveStatus"）实现；然后将取得的 VavleStatus 的当前值取反（当前值为，取反后为0；当前值为0，取反后为1），由运算符“!”实现；最后将取反后的值赋给二进制变量“temp”进行暂存，由函数 SetTagBit 实现。

（3）组态注意事项

- 二进制变量 ValveStatus 在实际应用中一般为外部变量，在定义其地址时不能是 PLC 的输入点（例如 I0.0），可以与 PLC 的位存储器对应（例如 M0.0）。
- 中间变量 temp 组态成内部变量即可。
- 将“画面窗口”的“调整大小”属性设置为“是”，可以在项目运行时让“画面窗口”的几何尺寸调整到与所调用的画面的几何尺寸一致。

**2. 没有确认的二元开关**

（1）任务定义

如图 4-84 所示，在过程画面中通过鼠标单击一个按钮（图中带黑色箭头的按钮）打开阀的操作面板（画面 VavlePanel. pdl 在画面窗口中显示），操作面板包含一个按钮（标识为“On/Off”的按钮）用来打开和关闭阀，单击标识“OK”的按钮可以关闭操作面板。执行打开阀的动作后，过程画面中的静态文本对象显示文本“阀打开”，同时圆背景闪烁；执行关闭阀的动作后，过程画面中的静态文本对象显示文本“阀关闭”，同时圆结束背景闪烁。



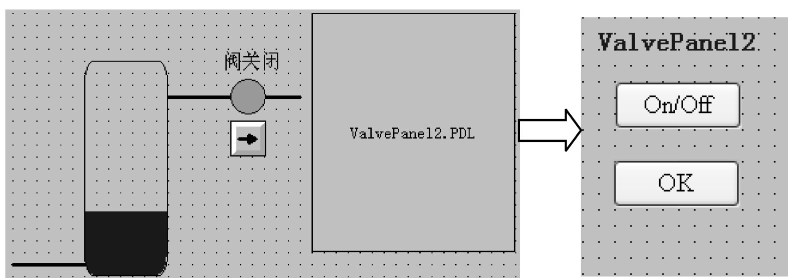



图 4-84 “没有确认的二元开关”组态任务定义

（2）组态步骤

1）组态二进制变量。在 WinCC 变量管理器中创建 1 个二进制类型的变量 ValveStatus ValveStatus 反映阀的当前状态（1：打开，0：关闭）。

2）组态操作面板。在图形编辑器中新建一个画面，保存画面文件为“ValvePanel2. PDL”；添加 WinCC 对象选项板中的“标准对象”→“静态文本”，将属性“字体”→“文本”设置为“ValvePanel2”；添加 2 个 WinCC 对象选项板中的“窗口对象”→“按钮”，对象名分别为“按钮 1”和“按钮 2”，显示文本分别为“On/Off”和“OK”；在“按钮 1”的事件“鼠标”→“按左键”处为“按钮 1”组态 1 个 C 动作，动作代码请参考下文；在“按钮 2”的事件“鼠标”→“按左键”处为“按钮 2”组态 1 个直接连接，将“来源”→“常数”→0 与“目标”→“当前窗口”→“显示”属性建立连接。

3）组态过程画面中的对象。与“有确认的二元开关”的组态步骤 2）相同，只是将“画面窗口”的“画面名称”属性设置为“ValvePanel2. PDL”，在此不再赘述。

4）保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

“按钮 1”的 C 动作代码如下：

```
{
    SetTagBit("ValveStatus",(short)! GetTagBit("ValveStatus"));
}
```

“按钮 1”的 C 动作代码的执行过程：取二进制变量 VavleStatus 的当前值，即阀的当前状态，由 GetTagBit（"ValveStatus"）实现；然后将取得的 VavleStatus 的当前值取反（当前值为，取反后为0；当前值为0，取反后为1），由运算符“!”实现；最后将取反后的值再赋给二进制变量 VavleStatus，由函数 SetTagBit 实现。

（3）组态注意事项

- 如果与“有确认的二元开关”在同一个项目中组态，则应将步骤 1）中组态的二进制变量的变量名进行更改。
- “没有确认的二元开关”与“有确认的二元开关”在组态上的最大差异体现在操作面板的“按钮 1”的 C 动作，“有确认的二元开关”将取反后的结果赋给 1 个中间变量 temp 进行暂存，而“没有确认的二元开关”将取反后的结果直接赋给反映阀的当前状态的二进制变量 ValveStatus。

### 4.7.6　颜色更改动态的组态方法

在组态 WinCC 过程画面动态化的过程中，颜色更改动态非常常见，例如，画面对象的背景色随变量值的改变发生动态变化，画面对象颜色的变化反映被监控对象的状态信息等。WinCC 可以采用多种方法组态对象的颜色更改动态，在介绍这些组态方法之前，先来了解 WinCC 系统中可组态的颜色，表 4-9 列出了 WinCC 系统中常用的 16 种基本颜色。

表 4-9　WinCC 系统常用的 16 种基本颜色

| 颜　　色 | 颜色值(十六进制) | 符号常数 |
|---|---|---|
| 红色 | 0x000000FF | CO_RED |
| 暗红色 | 0x00000080 | CO_DKRED |
| 绿色 | 0x0000FF00 | CO_GREEN |
| 暗绿色 | 0x00008000 | CO_DKGREEN |
| 蓝色 | 0x00FF0000 | CO_BLUE |
| 暗蓝色 | 0x00800000 | CO_DKBLUE |
| 青色 | 0x00FFFF00 | CO_CYAN |
| 暗青色 | 0x00808000 | CO_DKCYAN |
| 黄色 | 0x0000FFFF | CO_YELLOW |
| 暗黄色 | 0x00008080 | CO_DKYELLOW |
| 绛红色 | 0x00FF00FF | CO_MAGENTA |
| 暗绛红色 | 0x00800080 | CO_DKMAGENTA |
| 淡灰色 | 0x00C000C0 | CO_LTGRAY |
| 灰色 | 0x00808080 | CO_DKGRAY |
| 黑色 | 0x00000000 | CO_BLACK |
| 白色 | 0x00FFFFFF | CO_WHITE |

在使用颜色符号常数时，要用#define 进行外部预定义。除了使用常用的 16 种基本颜色，还可以使用调色板颜色的中间值来产生混合色。

组态颜色更改的动态，WinCC 系统可采用如下方法：

- 动态向导的“基于表格的颜色动态”功能。
- 动态对话框。
- C 动作。
- VBS 动作。

下面详细介绍这些颜色更改动态的组态方法。

**1. 动态向导——基于表格的颜色动态**

画面对象的颜色属性在 WinCC 项目运行时可使用表格进行控制。为此，可为每个表格条目分配一种颜色和一个逻辑表达式，其中逻辑表达式由来自 WinCC 变量管理器中一个或多个 WinCC 变量的位信息组合而成。表格最多可以有 10 个条目，即最多可以定义 10



种颜色。随后将对表格条目进行优先次序的区分，第一个表格条目具有最高的优先级，而最后一个表格条目具有最低的优先级。如果在项目运行期间产生一个以上的表达式，则将使用具有最高优先级的表达式。

（1）基于表格的颜色动态的组态对话框

组态基于表格的颜色动态的对话框如图 4-85 所示，整个对话框分为两个部分：颜色表区（图 4-85 中名称为“Table for 圆 1”的框内）和表达式编辑区（图 4-85 中名称为“Expression”的框内）。

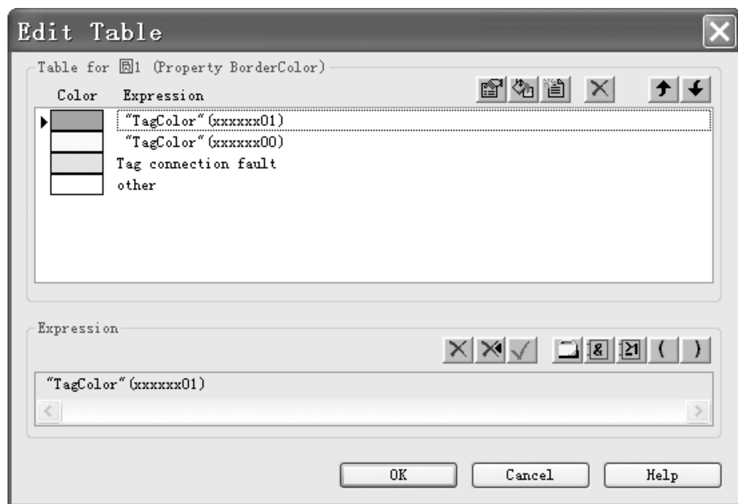



图 4-85 基于表格的颜色动态特性组态对话框

颜色表列出了每个表格条目分配的颜色（“Color”列）和对应的逻辑表达式（“Expression”列），提供了以下几个可供操作的按钮。

- “属性”按钮：所选行中的所有位标记均按次序显示，并可对其进行修改。
- “颜色”按钮：将打开“颜色”对话框，也可以通过双击所选择的行来打开“颜色”对话框。
- “新建列表条目”按钮：将新的行插入到表格中。
- “删除高亮显示的条目”按钮：高亮显示的行将删除。
- “向上”按钮：所选行在表格中将向上移动，从而给予了较高的优先级。
- “向下”按钮：所选行在表格中将向下移动，从而给予了较低的优先级。

表达式编辑区用于编辑为颜色表条目分配的表达式，不能手动编辑表达式，但提供了以下几个可供操作的按钮：

- “删除整个表达式”按钮：删除整个表达式。
- “删除表达式的最后部分”按钮：删除表达式的最后部分。
- “接受表格中的表达式”按钮：将表达式传送给高亮显示的表格。
- “变量选择”按钮：打开“变量选择”对话框，如图 4-86 所示。所有位、字节、字和双字类型的变量都可以显示。当已经选择了变量时，变量的位标记将显示，待检查的位将在变量位标记中进行设置。

- “与”按钮：如果句法上允许，则逻辑“与”运算将添加到表达式中。
- “或”按钮：如果句法上允许，则逻辑“或”运算将添加到表达式中。
- “开括号”按钮：如果句法上允许，则开括号将添加到表达式中。
- “闭括号”按钮：如果句法上允许，则闭括号将添加到表达式中。

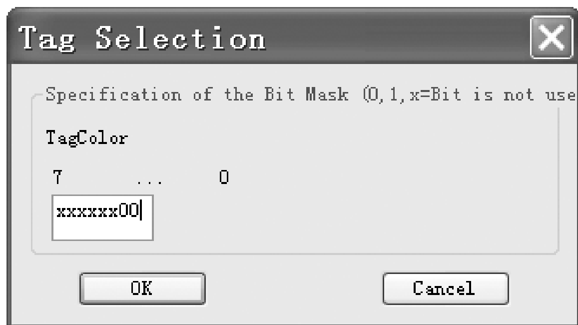



图 4-86　变量选择对话框

在添加新的颜色更改条件之前，必须使用“新建列表条目”按钮插入一个新的表格条目。随后，空白条目将自动加亮，此时可以单击“颜色”按钮打开“颜色”对话框，设置该条目的颜色，并在表达式编辑区添加颜色更改的条件（即表达式），完成表达式编辑后可以单击“接受表格中的表达式”按钮，将表达式添加到当前加亮的表格条目中。

“变量选择”对话框用于输入要检查的变量位，最多可以显示 4 B（对二进制变量来说只有一个变量位）的变量位。在该对话框中输入要检查的状态。“x”表示该位可忽略。如果输入“0”或“1”，则相关变量位的状态将分别进行检查，如图 4-86 所示。

（2）基于表格的颜色动态应用实例

任务定义：如图 4-87 所示，矩形的背景色将根据“选项组”对象的不同选择更改为各种颜色，总共有 8 种可供选择的颜色。

组态方法和步骤：为了实现图 4-87 所示的动态效果，需要在过程画面中从 WinCC 对象选项板添加 1 个“标准对象”→“矩形”，1 个“窗口对象”→“选项组”，在 WinCC 变量管理器中创建 1 个类型为“无符号 8 位数”的变量，详细组态步骤如下：

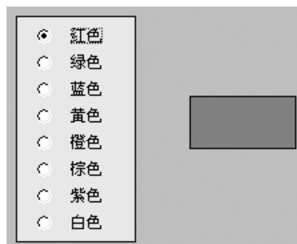



图 4-87　基于表格的颜色动态应用实例

1）组态变量。在 WinCC 变量管理器中创建 1 个类型为“无符号 8 位数”的变量 TagColor，用于和过程画面中添加的“选项组”建立变量连接。

2）组态过程画面中的对象。

- “选项组”对象：在过程画面中从 WinCC 对象选项板添加 1 个“窗口对象”→“选项组”。打开其“对象属性”对话框，将属性“几何”→“框数量”设置为 8；属性“字体”→“索引”为 1 时，设置属性“字体”→“文本”为“红色”，属性“字体”→“索引”为 2 时，设置属性“字体”→“文本”为“绿色”，依此方法按照图 4-87 中设置“选项组”的显示文本；在属性“输出/输入”→“选择



框”的“动态”列为“选项组”对象建立与变量 TagColor 的变量连接，并将对应的“更新周期”列设置为“有变化时”。

- “矩形对象”：在过程画面中从 WinCC 对象选项板添加 1 个“标准对象”→“矩形”，无需设置其属性。

3）组态颜色更改的表格。选中过程画面中已经添加的“矩形”对象，用鼠标双击“动态向导”窗口（“动态向导”窗口的使用方法请参考本章4.4.2 节的内容）的“标准动态”→“基于表格的颜色动态”，打开“欢迎来到动态向导”对话框，开始“基于表格的颜色动态”的动态向导过程；单击“下一步”按钮，打开“设置选项”对话框，在“当前对象的属性”下拉列表框中选择“BackColor”属性；单击“下一步”按钮，在接下来的对话框中选择“有变化时”作为触发器；单击“下一步”按钮，打开“编辑表格”对话框，即图 4-85 所示的对话框，在此对话框中按照上面介绍的方法组态“矩形”对象的颜色更改动态表格，组态结果如图 4-88 所示；单击“OK”按钮，完成动态向导。

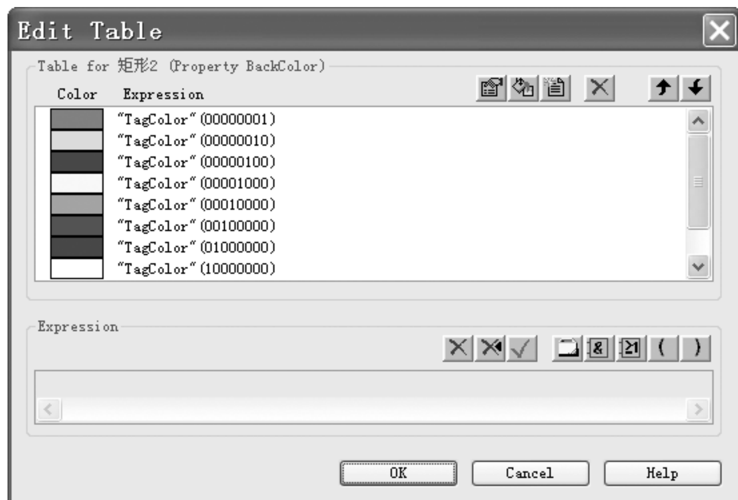



图 4-88 “矩形”对象的颜色更改动态表格

4）保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

**2. 动态对话框组态颜色更改动态**

本章 4.4.4 节详细介绍了动态对话框的概念、动态对话框组态窗口的使用方法以及应用实例，这里从颜色更改动态方面的应用再来熟悉动态对话框的灵活运用。

（1）任务定义

工件的位置将影响工件的背景色，在实例中，由矩形来模拟工件，由内部变量来模拟工件的位置，通过 I/O 域可以对位置进行修改。

（2）组态方法与步骤

根据任务定义，需要组态一个类型为“无符号 16 位数”的内部变量 TagPosition 用于模拟工件的位置；需要将“I/O”域插入到过程画面，并将其与变量 TagPosition 建立变量连接，这样可以通过“I/O”域修改工件的位置；需要将“矩形”插入到过程画面，并通过与变量 TagPosition 的变量连接对“矩形”的“位置 X”属性进行动态化。详细组态步

骤如下：

1）组态变量。在 WinCC 变量管理器中创建 1 个类型为“无符号 16 位数”的变量 TagPosition，用于模拟工件的位置，在该变量的“变量属性”对话框的“限制/报告”选项卡中将 TagPosition 的下限设为 0，上限设为 1000。

2）组态过程画面中的对象。将 1 个对象选项板中的“智能对象”→“I/O 域”插入到画面中，并在弹出的组态对话框中将“I/O 域”与变量 TagPoistion 建立连接；将 1 个对象选项板中的“标准对象”→“矩形”插入到画面中，打开其“对象属性”窗口，在属性“几何”→“位置 X”的“动态”列组态一个与变量 TagPosition 的变量连接，并将对应的“更新周期”列设置为“有变化时”。

3）用动态对话框组态“矩形”的“背景颜色”更改动态。在“矩形”的属性“颜色”→“背景颜色”的“动态”列的白色灯泡处用鼠标右键单击，在弹出的下拉菜单中选择“动态对话框”，打开“动态值范围”对话框，如图 4-89 所示，在“表达式/公式”选择了变量 TagPosition，并在“表达式/公式的结果”区域添加了 3 个数值范围，即 0≤TagPosition≤100 时工件背景色为绿色，100 < TagPosition≤300 时工件背景色为黄色，300 < TagPosition≤500 时工件背景为红色，其他位置（即 TagPosition > 500）时为灰色。




图 4-89　工件背景色更改动态的动态对话框设置

4）保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

### 3. C 动作组态颜色更改动态

用 C 动作组态颜色更改动态，虽然较动态向导和动态对话框方法烦琐，但具有更大的灵活性，能解决更为复杂的问题。这里通过一个实例来介绍 C 动作在颜色更改动态方面的应用。

（1）任务定义

随着变量值的增加，画面对象颜色将逐渐由深变浅。

（2）组态方法和步骤

为了实现上述任务，使用 1 个对象选项板中的“标准对象”→“圆”，其颜色随变量值的变化而变化，为了输入变量值，使用 1 个对象选项板中的“窗口对象”→“滚动条对象”。详细组态步骤如下：



1）组态变量。在 WinCC 变量管理器中创建 1 个类型为“无符号 16 位数”的内部变量 TagColorCAction。

2）组态画面中的对象。将 1 个对象选项板中的“窗口对象”→“滚动条对象”添加到过程画面，并在弹出的“滚动条组态”窗口中将其与变量 TagColorCAction 建立连接，并将“更新”设为“有变化时”，最大值设为 255，最小值设为 0，步长设为 10；将 1 个对象选项板中的“标准对象”→“圆”添加到过程画面，打开其“对象属性”窗口，在属性“颜色”→“背景颜色”的“动态”列的白色灯泡上单击鼠标右键，在弹出的菜单中选择“C 动作”，打开“编辑动作”窗口，组态一个 C 动作，用于根据变量 TagColor-CAction 提供颜色值，一旦该变量值发生改变就触发这个动作。动作代码如下：

```
{
    return (GetTagDWord("TagColorCAction") < <8);
}
```

3）保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

这里组态的 C 动作将读取的变量 TagColorCAction 向左移动 8 位后作为返回值为“圆”的“背景颜色”属性提供颜色值。

颜色值通过指定红、绿、蓝的值来进行编码，在 24 位颜色值中为每种颜色保留 8 位。在本实例中，变量值向左移动了 8 位，因此代表绿色，即“圆”背景色随变量 TagColor-CAction 的值增大由最初的黑色逐渐变为绿色。如果左移 16 位，则颜色应该从黑色变为蓝色，而左移 24 位是从黑色变为红色。

#### 4. VBS 动作组态颜色更改动态

和 C 动作一样，VBS 动作组态颜色更改动态具有很大的灵活性，能解决比较复杂的问题。这里同样通过一个实例来介绍 VBS 动作在颜色更改动态方面的应用。

（1）任务分析

一个酿酒厂的酿造过程的温度将要以图形方式显示在棒图上，一个趋势指示温度值是上升还是下降，温度显示范围是 0 ~ 100℃，棒图颜色将随温度的升高发生改变：0 ~ 75℃时棒图颜色为绿色，76 ~ 90℃时为黄色，高于 90℃时为红色。

（2）组态方法与步骤

为了实现以上任务，使用 1 个对象选项板中的“智能对象”→“棒图”，用于图形方式显示酿造过程的温度；使用 1 个对象选项板中的“窗口对象”→“滚动条”，用于模拟温度的变化。详细组态步骤如下：

1）组态变量。在 WinCC 变量管理器中创建 1 个类型为“无符号 8 位数”的内部变量 TagTemperature。

2）组态画面中的对象。

- 将 1 个对象选项板中的“窗口对象”→“滚动条对象”添加到过程画面，并在弹出的“滚动条组态”窗口中将其与变量 TagTemperature 建立连接，并将“更新”设为“有变化时”，最大值设为 100，最小值设为 0，步长设为 5。
- 将 1 个对象选项板中的“智能对象”→“棒图”添加到过程画面，在弹出的“棒图组态”对话框中将其与变量 TagTemperature 建立连接，并将“更新”设为“有

变化时”，最大值设为 100，最小值设为 0；打开“棒图”的“对象属性”对话框，将属性“其他”→“趋势”设为“是”，用于指示温度是上升还是下降的趋势；在属性“颜色”→“棒图颜色”的“动态”列的白色灯泡上单击鼠标右键，在弹出的菜单中选择“VBS 动作”，打开“编辑 VB 动作”窗口，组态 1 个 VBS 动作，用于根据变量 TagTemperature 的“棒图颜色”更改动态，一旦该变量值发生改变就触发这个动作。动作代码如下：

```
Function BackColor2_Trigger( ByVal Item)
    Dim objBar
    Dim MyTag
    Set MyTag = HMIRuntime. Tags( "TagTemperature" )
    MyTag. Read
    Set objBar = ScreenItems( "棒图 2" )
    If ( MyTag. Value  > 90 ) Then
        objBar. BackColor2 = vbRed
    Elseif ( ( MyTag. Value  > = 75 ) ) Then
        objBar. BackColor2 = vbYellow
    Else
        objBar. BackColor2 = vbGreen
    End IF
End Function
```

3）保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

将棒图的“趋势”属性设为“是”，在“棒图”的左下角将出现一个三角形的趋势标识，可以指示棒图显示的值当前是上升还是下降的趋势。

在本实例中，VBS 动作的颜色值使用了 VBS 常量 vbRed（红色）、vbGreen（绿色）和 vbYellow（黄色）来定义。除了使用 VBS 常量，还可以用函数 RGB（R，G，B）来定义颜色值，参数 R、G 和 B 分别代表了红色、绿色和蓝色的颜色值，取值范围 0～255。因此，可以用 RGB（255，0，0）定义红色，RGB（255，255，0）定义黄色，RGB（0，255，0）定义绿色等。

### 4.7.7　移动过程动画的组态方法

在组态过程画面动态化的过程中，有时需要组态一些画面对象的移动过程动画，例如根据变量值将画面对象移动到屏幕上的指定位置，画面对象在屏幕上转动一定的角度等。WinCC 组态画面对象的移动过程动画可以通过两种方式实现，一种方式是改变画面对象的几何属性，例如“位置 X”、“位置 Y”和“旋转角”等属性，另一种方式是将画面对象在不同位置的状态保存在画面文件中，通过更新画面的显示来实现移动过程动画。第一种方式一般将画面对象的几何属性与变量建立连接，改变变量值即可移动画面对象。第二种方式需要借助对象选项板中的“智能对象”→“状态显示”对象实现。

下面通过一些应用实例来介绍两种方式实现移动过程动画的组态方法。



**1. 使用动态向导组态移动过程的动画**

（1）任务定义

变量值改变时对象的屏幕位置随之改变。X 和 Y 轴位置使用不同的变量，通过动态向导方法执行组态。

（2）组态方法与步骤

为了实现上述任务，可以使用 1 个对象选项板中的“标准对象”→“圆”在屏幕上移动，使用 2 个对象选项板中的“窗口对象”→“滚动条对象”用于变量输入。具体组态步骤如下：

1）组态变量。在 WinCC 变量管理器中创建 2 个类型为“无符号 32 位数”的变量 TagPositionX 和 TagPositionY。

2）组态画面对象。添加 2 个对象选项板中的“窗口对象”→“滚动条对象”到画面内，在各自弹出的“滚动条组态”对话框中，将它们分别与变量 TagPositionX 和 TagPositionY 建立变量连接，并将最大值设置为 255；添加 1 个对象选项板中的“标准对象”→“圆”到画面内。

3）组态动态向导。在“圆”处于选中状态时，用鼠标双击“动态向导”窗口的“标准动态”→“移动对象”，开始动态向导过程；在初始弹出的对话框中单击“下一步”按钮，在弹出的“选择触发器”对话框中选择“变量”；单击“下一步”按钮，在弹出的“设置选项”对话框中将“X 方向变量名”设为 TagPositionX，“Y 方向变量名”设为 TagPositionY，分别输入 0 和 255 作为规范化的下限和上限；单击“下一步”按钮，在接下来的对话框中指定“圆”在屏幕上的移动区域；单击“完成”按钮结束动态向导。

4）打开“圆”的“对象属性”窗口，将属性“几何”→“位置 X”和属性“几何”→“位置 Y”的由动态向导生成的 C 动作的变量触发器设置为“有变化时”。

5）保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

本实例属于第一种方式实现移动过程的动画，通过动态向导方法组态了画面对象的几何属性与变量的连接，变量的值通过滚动条对象输入。第一种方式实现移动过程动画原理上基本一致，即将画面对象的几何属性与变量建立连接，可变的是变量值的输入方法。变量值的输入既可以像本实例一样通过滚动条、I/O 域等对象手动输入，也可以通过周期触发的动作自动输入，这样的动作可以采用动态对话框、C 动作和 VBS 动作等方法组态。

**2. 使用状态显示组态移动过程的动画**

状态显示在对象选项板的“智能对象”区，能提供几乎任何数量的不同状态的显示。状态通过变量来定义，变量的值对应于各个状态，状态通过分配的画面显示。状态显示对象在运行系统中的大小和属性在图形编辑器中进行组态。

（1）“状态显示组态”对话框

状态显示对象在画面中添加后会自动弹出“状态显示组态”对话框，如图 4-90 所示，“状态显示组态”对话框允许快速组态状态显示的重要属性。组态对话框也可以用对象的弹出式菜单打开，特定的属性可以在“对象属性”窗口内更改。以下是“状态显示组态”对话框各部分功能以及组态方法的详细介绍。

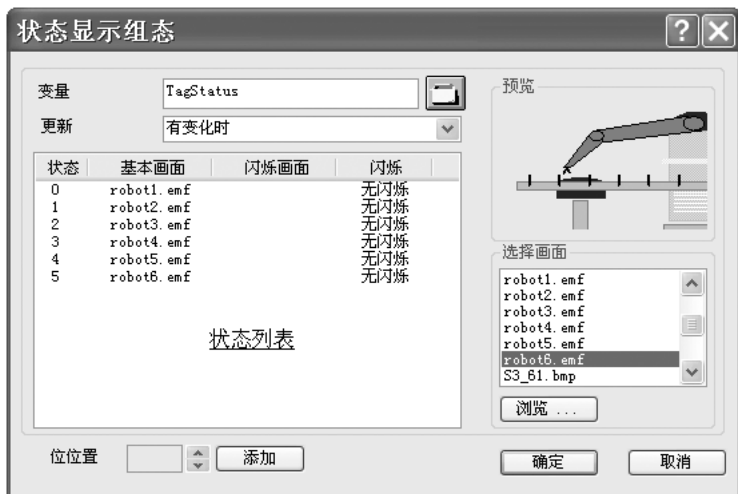



图 4-90　“状态显示”组态对话框

- 变量：状态显示对象希望显示的状态可以通过嵌入变量动态地设置。输入期望的变量名称，或单击按钮以打开“变量 – 项目：...”窗口来选择变量。
- 更新：组态状态显示对象更新显示状态的频率。可以输入期望的更新频率或从下拉式列表中进行选择。
- 状态列表：列出了所有已组态的状态所链接的画面和闪烁频率。状态列表中可单击鼠标右键弹出菜单来插入状态和更改各项设置。状态所链接的画面通过鼠标从“选择画面”框中拖动到状态列表中所需位置。
- 位位置：状态也可以分配给变量的特定位位置。单击箭头键来选择位位置。单击“添加”按钮以在状态列表中为此位位置输入状态。再次单击“添加”按钮以添加下一个最高的位位置。
- 预览：对“选择画面”框中选择的画面作预览显示。
- 选择画面：选择画面框列出了所有在当前 WinCC 项目的图形目录“GraCS”中的画面。可以插入下列格式的图形和画面：BMP、DIB、ICO、CUR、EMF、WMF、GIF 和 JPG。用鼠标将选择画面框中的画面拖动到状态列表中期望的位置，以便分配其状态为基准画面或闪烁画面。单击“浏览 ...”按钮插入来自选择画面框的附加画面。

（2）应用实例

1）任务定义：如图 4-91 所示，robot1.emf ~ robot6.emf 六幅图片显示了一个机械手的 6 种状态，按顺序更新显示这六幅图片，可以看到机械手的一个完整的工作过程。要求在 WinCC 运行系统的过程画面中演示这个机械手工作过程的动画。

2）任务分析：要完成以上任务，需要在过程画面中添加 1 个对象选项板中的“智能对象”→“状态显示”对象来更新显示机械手的 6 个状态画面；需要在 WinCC 变量管理器中创建 1 个内部变量定义机械手的状态；为了自动输入和显示变量的值，在画面中添加 1 个对象选项板中的“智能对象”→“输入/输出域”对象。

3）组态方法与步骤：



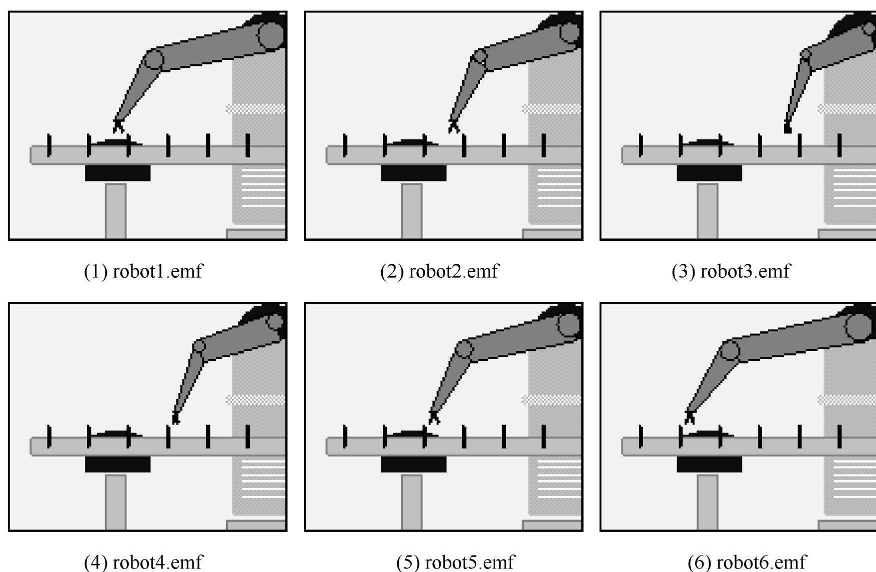



图 4-91　状态显示应用实例示意图

① 组态变量。在变量管理器中创建 1 个类型为“无符号 8 位数”的内部变量 TagRobot，用于定义机械手的状态。

② 组态画面对象。将 1 个对象选项板中的“智能对象”→“状态显示”对象添加到画面中，在弹出的“状态显示组态”对话框进行设置：在“变量”框中输入或单击按钮以打开“变量 - 项目：...”窗口选择变量 TagRobot；将“更新”设置为“有变化时”，画面的更新显示将在变量 TagRobot 的值变化时进行；在“状态列表”中单击鼠标右键，在弹出的菜单中选择“添加”，将在“状态列表”中新增 1 个状态行，将“选择画面”框中的对应画面文件用鼠标拖动到该状态行，整个对话框的组态如图 4-92 所示。将 1 个对象选项板的“智能对象”→“输入/输出域”对象添加到画面中，打开其“对象属性”对话框，在属性“输出/输入”→“输出值”的动态列组态 1 个“动态对话框”：选择“数据类型”为“直接”，在“表达式/公式”框中输入 TagRobot%6 +1，将触发器设为“标准周期”→“500 ms”；在事件“输出/输入”→“更改”的动作处组态一个直接连接：将来源“属性”→“该对象”→“输出值”与目标“变量”→TagRobot 建立直接连接。

③ 保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

在本实例中，“表达式/公式”TagRobot%6 +1 的取值范围是 1 ~6，用一个 500 ms 的标准周期触发的动态和直接连接来自动更新表达式的值，值得注意的是在画面中过多使用周期触发的动态可能在项目运行时因 CPU 负载过重导致系统运行不畅。

设置“状态显示组态”对话框前，应将需要添加的画面文件复制到当前 WinCC 项目的图形目录“GraCS”中，这样才能在“状态显示组态”对话框的“选择画面”框中显示。

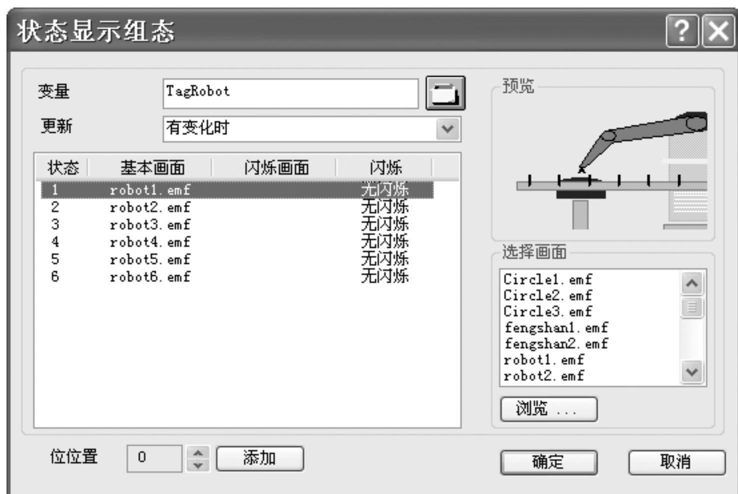



图 4-92　应用实例的“状态显示组态”对话框设置

### 4.7.8　复选框应用于位操作的组态方法

**1. 复选框的功能**

复选框在对象选项板的“窗口对象”区，是 Windows 应用程序常用的控件。复选框允许操作员选择多个选项并且选中相应的复选框。在默认状态下可激活复选框，这样用户只要按需改变默认数值。

在 WinCC 项目中，复选框可以用来对变量的位进行操作，将复选框与变量建立连接，在运行状态下选中复选框的某个框，将对变量相应的位进行置位，相反则进行复位。复选框的这种置位和复位操作可以用来控制对象的显示和隐藏，控制对象的启动和停止，例如可用做阀控制器和电机控制器等。下面用一个实例来介绍复选框的组态方法。

**2. 应用实例**

（1）任务定义

根据变量中指定位的置位/复位来显示和隐藏对象。

（2）组态方法与步骤

可采用一个对象选项板中的“窗口对象”→“复选框”来实现，它可以对变量的各个位进行置位和复位操作。多个对象选项板中的“标准对象”→“多边形”根据这些位的状态定义其显示或隐藏的状态（1：显示，0：隐藏）。详细组态步骤如下：

1）组态变量。在变量管理器中创建 1 个类型为“无符号 8 位数”的变量，变量名称为 TagShow。

2）组态画面对象。在画面中添加 1 个对象选项板中的“窗口对象”→“复选框”，打开其“对象属性”窗口，将属性“几何”→“框数量”设置为 7，在属性“字体”→“索引”和属性“字体”→“文本”处，为每个索引值（索引值对应方框在复选框中的位置）输入要由相应位进行切换的对象名称，在属性“输出/输入”→“选择框”的动态处组态一个复选框与变量 TagShow 的变量连接，并将对应的更新周期设为变量“有变化时”。在画面中添加 7 个对象选项板中的“标准对象”→“多边形”，对象名称为多边



形 1 ~ 多边形 7，如图 4-93 左图所示。选中多边形 1，在属性“其他”→“显示”的动态处组态一个动态对话框，选择“数据类型”为“位”，在“变量 - 位”框中单击变量选择按钮（左边按钮）选择变量 TagShow，单击位选择按钮（右边按钮）打开“位选择”对话框，并选择第 1 个位（Bit0），如图 4-93 右图所示，在“表达式/公式的结果”框中设置“置位”→“是”，“未置位”→“否”，将触发器设置为变量 TagShow“有变化时”。对其余 6 个多边形按同样的方法进行处理，但需在“位选择”对话框中修改每个对象对应的位编号，例如多边形 2 对应 Bit1，多边形 3 对应 Bit2，以此类推。




图 4-93　复选框应用实例组态示意图

3）保存所有组态，运行 WinCC 项目测试组态结果是否与任务定义一致。

本实例中，复选框与 1 个类型为“无符号 8 位数”的变量建立了连接，此时复选框可以有 8 个框与变量的 8 个数据位对应，即索引号为 1 的框与变量的最低位 Bit0 对应，以此类推。在运行状态下，选中复选框的某个框，即对与其对应的数据位置位，反之则复位。复选框最多可以有 32 个框，即最多可以与类型为“32 位无符号数”或“32 位有符号数”建立连接。

# 第 5 章

# 消 息 系 统

消息系统是 SCADA 系统中非常重要的组成部分，用来监视自动化系统的异常和事件。借助于可视或有声提示，消息系统可以帮助操作人员快速发现、定位并解决处理自动化系统的故障和错误，从而有效地减少停机事件和生产事故的发生。通过消息归档，消息系统可以追溯报警和事件，进而分析事故起因、认定事故责任等。

本章将介绍 WinCC 的消息系统的基本概念、组态界面以及组态消息显示和消息归档的方法和技巧。

## 5.1　WinCC 消息系统的相关概念

本节将介绍消息系统的一些相关概念，为熟悉和掌握消息系统的组态打下基础。

### 5.1.1　消息和报警

1. WinCC 的消息源

WinCC 消息系统的消息源可以分为两大类：由 WinCC 数据管理器的消息变量触发的消息和来自于 WinCC 系统内部的消息，如图 5-1 所示。

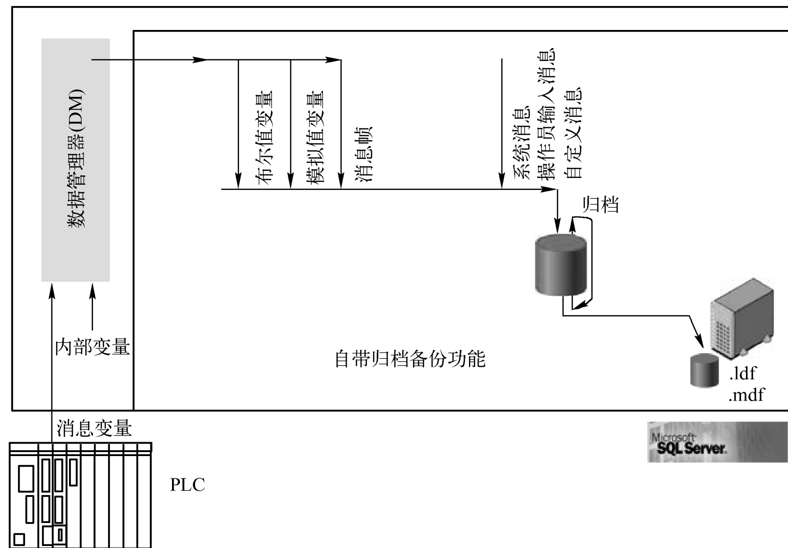



图 5-1　WinCC 消息系统的消息源



对于来自数据管理器的消息变量，又可以分为三种：布尔值变量、模拟量变量（上下限报警）和消息帧。其中，布尔值变量和模拟量变量较常用，分别是指 WinCC 普通的位报警和模拟量上下限报警。而消息帧是 WinCC 使用一个 RawData（原始数据类型）变量获取 PLC 主动发送的所有消息帧，然后用特定格式的动态链接库从该数据帧中提取出相关的报警信息，在 HMI 上实现归档显示等功能。

**2. 事件**

消息系统分为二进制事件和监控事件两种事件：

- 二进制事件是内部或外部变量的状态改变。
- WinCC 报警记录不直接支持监控事件。监控事件包括：归档和只读存储器的溢出、打印机消息、服务器故障以及过程通信中断。

**3. 消息的分类**

消息系统系统的目的是在一个重要点按时间顺序标记和归档事件，该点在消息处理过程中是偶发的。事件或消息帧可以引发消息。消息系统分为三种消息。

- 运行消息：用于显示过程中的状态。
- 报警消息：用于显示过程中的错误。
- 系统消息：用来显示来自其他应用程序的错误。

在报警记录中，可以按消息类别和消息类型把具有类似特性的消息（例如具有相同的消息状态确认方法或颜色分配）编组在一起。

**4. 消息的类别和类型**

消息类别用于定义消息的多个基本设置。关于确认方法，各消息类别互不相同。在报警记录中，预组态以下消息类别：“错误”、“系统，需要确认”和“系统，没有确认”。具有相同确认方法的消息可以归入单个消息类别。WinCC 最多可定义 16 个消息类别。

消息类型为消息类别的子组，并可根据消息状态的颜色分配进行区分。最多可为每个消息类别创建 16 个消息类型。

**5. 初始值消息、新值消息**

“初始值消息”用于描述这样一种消息处理形式，在该形式下，自从前一次确认后消息列表中的第一条发生状态改变的消息被高亮显示。

“新值消息”用于描述这样一种消息处理形式，在该形式下，自从前一次确认后消息列表中发生状态改变的消息被高亮显示。

**6. 消息窗口**

在运行系统中，消息状态的变化被输出到消息窗口中显示。可在图形编辑器中组态消息窗口的外观和操作选项。

消息窗口以表格的形式包含了尚待显示的所有消息。每条要显示的消息均在它自己的行上出现，即在消息行中显示。

消息窗口的内容受用户自定义的过滤设置影响，例如，根据消息块进行排序。

根据消息窗口所显示的消息来源，可将消息窗口分为 6 种类型：

- 消息列表用来显示当前未决的消息。
- 短期归档列表用于显示存储在消息归档中的消息。当新的消息到达时，消息显示立

即更新。

- 长期归档列表用于显示存储在消息归档中的消息。
- 在锁定列表中显示已经在系统中锁定的所有消息。已锁定的消息可通过工具栏上的按钮解锁。
- 统计列表包含了与消息相关的统计信息。
- 已隐藏消息列表中显示由于自动或手动隐藏而无法在消息列表、短期归档列表或长期归档列表中看见的所有消息。

**7. 消息行**

在消息窗口中，每个消息均显示在自己的消息行中。消息行的内容取决于将要显示的消息块：

- 对于系统消息块，将显示消息块的值，例如日期和时间。
- 对于过程和用户文本块，将显示其内容，例如用户所定义的文本。

**8. 消息块**

在运行系统中，消息的状态改变显示在消息行中。用消息块定义要在消息行中显示的信息。有三种不同类型的消息块：

- 系统块（例如日期、时间、持续时间以及注释等）便于信息分类，它是预定义的，不可自由使用。有了系统块，消息块（例如：时间）的值显示在消息行中。
- 用户文本块允许给消息分配多达10个可自由定义的不同文本。有了用户文本块，消息块的内容，即定义的文本，将显示在消息行中。
- 通过使用过程值块，可在消息行中显示变量值。可定义变量值所使用的格式。有了过程值块，消息块的内容，即定义的变量值，将显示在消息行中。

**9. 单个消息和组消息**

WinCC 报警记录中，区分两种形式的消息：

- 单个消息：给每个事件都被分配一条单独的消息。
- 组消息：用于总结多个单个消息。结果是，与单个消息链接的事件只触发组消息。不能确定触发组消息的单个消息。可以为每个消息类别和消息类型设置一个组消息。此外，还可以用任何单个消息创建组消息。

**10. 消息的操作步骤**

报警记录支持多种消息操作步骤：

- 在位消息的操作步骤中，控制系统通过消息变量标记发生的事件。消息的时间标志（日期和时间）由报警记录标记。
- 按时间顺序排列的消息，在事件发生时，控制系统将发送带有消息数据的消息帧，报警记录将对此进行分析。消息的时间标志（日期和时间）由控件标记。
- 模拟量报警可用于监控模拟变量的变化，了解其超出上限和下限值的情况。

**11. 消息变量**

在位消息的操作步骤中，控制系统将通过消息变量标记过程中事件的发生。通过一个消息变量可以显示多条消息。

**12. 消息事件和消息状态**

消息事件就是消息的“到达”、“离开”和“确认”。所有消息事件都存储在消息归



档中。

消息状态就是消息的可能状态：“已到达”、“已离开” 和 “已确认”。

### 5.1.2 确认

对消息进行确认，就是定义在消息的“到达”和“离开”状态之间，如何在运行系统中显示和处理消息。

**1. 单个确认和组确认**

确认未决的消息，可使用下面两种方式中的一种：

- 必须单独确认未分配有“组确认”属性的消息。
- 组确认来集中确认所有具有“组确认”属性的单个消息，这些属性在消息窗口中均是可见的。

消息确认记录可包括确认用户和所使用计算机的身份。为此，需在 WinCC 报警控件中消息的消息行中插入“用户名”和“计算机名”系统块。在确认消息时，用户名和计算机名均不在到达消息的消息行中显示，而是在短期和长期归档列表中的操作员输入消息确认中显示。必须使用“操作员输入消息”按钮在 WinCC 报警控件中的“消息列表”标签上激活操作员输入消息。

**2. 确认方法**

确认方法就是从消息“到达”时起到消息“离开”时止的这段时间内，对消息进行显示和处理所采用的方法。在报警记录中，可以使用下列确认方法：

- 单个消息，不带确认。
- 单个消息，带“来”确认。
- 单个消息，带“去”确认。
- 初始值消息，带单个确认。
- 新值消息，带单个确认。
- 新值消息，带双确认。
- 消息，不带“离开”状态，不带确认。
- 消息，不带“离开”状态，带确认。

**3. 确认变量**

在确认变量中，存储消息的“确认状态”。因此，可使用确认变量控制中央信号设备。

**4. 状态变量**

在状态变量中，存储该消息类型的“到达/离开”状态以及要求确认的消息的标识符。

### 5.1.3 归档

根据消息类别，WinCC 可将消息状态中发生的变化写入到可组态的消息归档中。为此，设置了不同的归档参数，例如归档大小、时间范围、切换时间等。如果超出了所组态标准中的某个标准，则覆盖归档中的最早的消息。

**1. 消息系统归档的原理**

在进行消息的归档时，WinCC 使用可组态容量的周期性归档（环形缓冲器）。用户可决定组态时使用或不使用备份，归档文件总是存储在本地计算机的相关项目下。WinCC 的消息归档由多个单独的分段构成，整个或者每个分段的容量和时间都是可组态的。例如：消息归档每一周归档所有发生的消息，每天归档一个分段的消息，如果两个关键指标（容量和日期）超出，将发生归档总容量超出，则最早的消息（即最早的归档分段）将被删除。如果归档单独分段容量超出，则一个新的归档分段（ES）将被创建。如果在线组态消息时（如使用在线 Delta 装入）一个新的分段也会创建。WinCC 消息系统归档的原理示意图如图 5-2 所示。

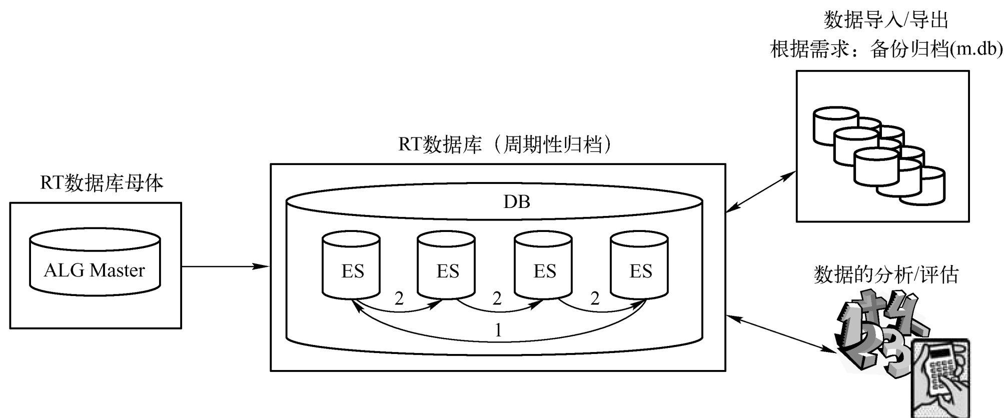



图 5-2　WinCC 消息系统的归档原理

归档数据库的备份可通过附加设置来指定。一条消息大概需要 120 B 的存储区。

**2. 消息系统的数据库**

运行数据分别存在数据库母体和几个运行数据库中。数据库母体管理运行数据库并以其为参考。数据库母体创建在项目的目录下，名字由项目名称和“RT”尾标构成。每个运行数据库包含特定时间段的归档数据，存储于项目目录的子目录“ArchiveManager/Alarm Logging”中。

运行数据库的名称构成：计算机名 + 项目名 + 数据类型的 ID（ALG = Alarm Logging，TLG_F = Tag Logging Fast，TLG_S = Tag Logging Slow） + 日期/时间（年、月、日、时、分）。即消息系统运行数据库报警记录分段名称为：计算机名_项目名_ALG_<起始时间>_<截止时间>. LDF 和计算机名_项目名_ALG_<起始时间>_<截止时间>. MDF。

如果片断的时间改变少于运行数据库的运行时间，将生成不完整的分段，为此将创建一个额外的文件，例如：SATCB_MyWinCC_ALG_200304300901. MDF，该文件的结构标明了消息被存储的时间。如果文件仅包括片断改变的时间则此期间没有消息被记录。

**3. 消息归档的分类**

消息归档分为长期归档和短期归档

● 短期归档：归档大小由要归档的消息数量限定。短期归档只能创建为连续归档。如



果已经达到归档消息的最大数量，将覆盖最早的消息。在短期归档中至多可存储 10 000 个消息状态改变。

- 长期归档：归档大小由要归档的消息时间跨度确定。长期归档可以是连续归档或顺序归档。在连续归档中，达到归档消息的最大数量以后，最早的消息将被覆盖。在顺序归档中，归档顺序进行，直到存储介质容量耗尽为止。

运行时，存储在短期归档和长期归档中的消息均可以在消息窗口中显示，而且短期归档和长期归档显示相同的归档数据。长期归档时可存储在线的注释，数据升级时，只有长期归档可以进行数据升级，短期的则丢失。

## 5.2 消息系统的组态

WinCC 消息系统包括组态系统和运行系统两个组件。消息系统的组态系统即报警记录编辑器，用于定义消息系统中的报警级别和类型、报警的内容和报警的时间等。使用报警记录编辑器可对报警消息进行组态，以便将其以期望的形式在消息系统的运行系统中显示。消息系统的运行系统主要负责过程值的监控、报警消息的显示、确认和管理等。

### 5.2.1 报警记录编辑器

报警记录编辑器是 WinCC 消息系统的组态系统，在 WinCC 项目管理器左侧浏览窗口中双击“报警记录”或鼠标右键单击“报警记录”，在弹出的菜单中选择“打开”，即可以进入报警记录编辑器，如图 5-3 所示。报警记录编辑器包含三个区域：

- 浏览窗口：要组态消息，按指定顺序访问树形视图中的文件夹使用弹出式菜单可访问单个区域及其元素。
- 数据窗口：显示可用对象的图标。双击对象，可访问相应的消息系统设置。可使用弹出式菜单来调用对象的属性，这些属性随选定的对象而不同。
- 表格窗口：包含一个具有所有已生成的单个消息和已组态的属性的表格。可通过鼠标双击，编辑表格的单个域。




图 5-3 报警记录编辑器

### 5.2.2　组态消息系统的基本步骤

消息系统的组态主要是指定哪些消息和内容将会显示在运行系统的消息窗口中，并进行归档。消息系统的组态分为以下步骤：

1）使用系统向导指定消息系统的基本设置。

2）按用户先决条件组态消息块。

3）组态消息类别和消息类型。

4）组态单个消息和组消息。

5）组态数据归档方法。

6）完成每一个组态步骤后，保存在报警记录中所做的更改。

### 5.2.3　启动报警记录的系统向导

系统向导用于预组态消息系统。频繁使用的消息块、消息类型和消息归档在预组态过程中创建，随后可按需要修改由向导创建的对象。使用系统向导预组态消息系统的步骤如下：

1）在报警记录编辑器中，通过菜单“文件”→“选择向导”或单击工具栏上的“选择向导”按钮，打开“选择向导”对话框，如图 5-4 所示，选择“系统向导”，然后单击“确定”按钮开始系统向导过程。

2）在打开的“系统向导”对话框中，单击“下一步”按钮，打开“系统向导：选择消息块”对话框，在这里指定哪个消息块将随系统向导创建，如图 5-5 所示。

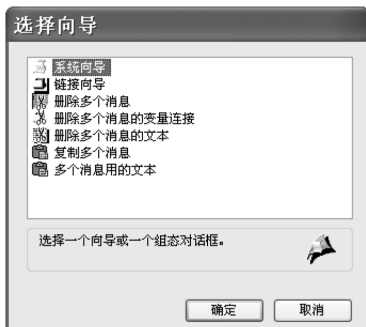



图 5-4　“选择向导”对话框

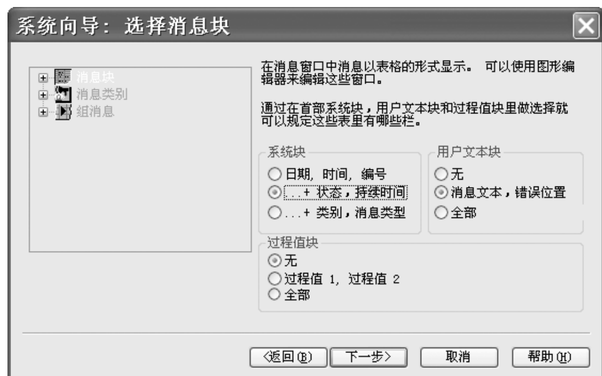



图 5-5　“系统向导：选择消息块”对话框

3）单击“下一步”按钮，打开“系统向导：预设置类别”对话框，用于指定消息类别及其相应的确认方法和相关的消息类型，有两种消息类别可供选择，如图 5-6 所示。

4）单击“下一步”按钮，进入“系统向导：结束”对话框，该对话框提供了将由系统向导创建的消息块和消息类别的概要信息，单击“完成”按钮结束向导过程。



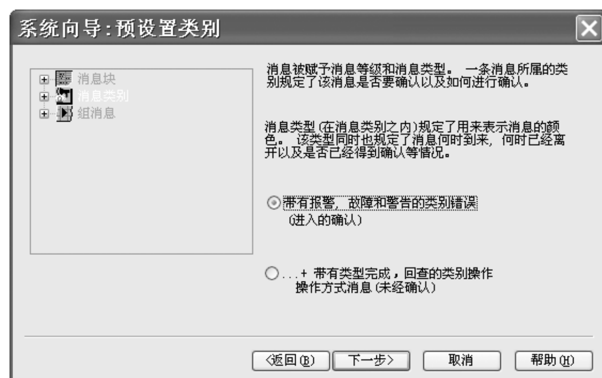



图 5–6 “系统向导：预设置类别”对话框

注意：如果再次运行系统向导，将丢失在上次运行系统向导后改变的所有设置。

## 5.2.4 组态消息块

消息的内容由消息块组成，每个消息块对应于消息窗口的表格显示中的某一列。前文已述及，存在有三组消息块：

- 具有系统数据的系统块，例如，日期、时间、消息号和状态。系统块的描述提供了有关单个系统块的信息。
- 具有说明性文本的用户文本块，例如，包含有故障原因或故障位置等信息的消息文本。每个消息系统允许有多达 10 个可组态的用户文本块。
- 用于将消息与过程值链接的过程值块，例如当前的填充量、温度或速度。对于每一个消息系统，每个消息系统允许有多达 10 个可组态的过程值块。

### 1. 消息块的添加/删除

频繁使用的消息块可以通过系统向导创建，通常情况下也可以不通过系统向导创建消息块或在系统向导创建的消息块的基础上进行修改和调整，例如，可以通过两种方法添加或删除消息块。

- 第一种方法：在报警记录编辑器的浏览窗口用鼠标右键单击“消息块”，或在浏览窗口选中“消息块”，然后在数据窗口的空白处单击鼠标右键，都可以弹出快捷菜单，在菜单中选择“消息块…”，打开“组态消息块”对话框，如图 5–7 所示。单击“添加”按钮可以打开“添加系统块”对话框，对话框显示的内容随“消息系统中的可用消息块”框中选择的消息块类型而变化。单击“删除”按钮可以删除所选的消息块。
- 第二种方法：在报警记录编辑器的浏览窗口中打开“消息块”的 + 号，打开消息块的对象结构。选中需要添加或删除的消息块类型，例如“系统块”，在数据窗口的空白处单击鼠标右键，在弹出的快捷菜单中选择“添加/删除”，打开“添加系统块”对话框，如图 5–8 所示。

在图 5–8 所示“添加系统块”对话框的“可用的系统块”列表内选中希望添加的块，单击[ -> ]按钮即可将其添加。同样，在“所选系统块”列表内选中希望删除的块，单击[ <- ]按钮即可将其删除。单击[ ->> ]按钮，可以添加所有可用的系统块，而单击[ <<- ]按钮，可以删除所有在“所选系统块”列表框中添加的系统块。以相同的方式添加/删除用户文本块和过程值块。

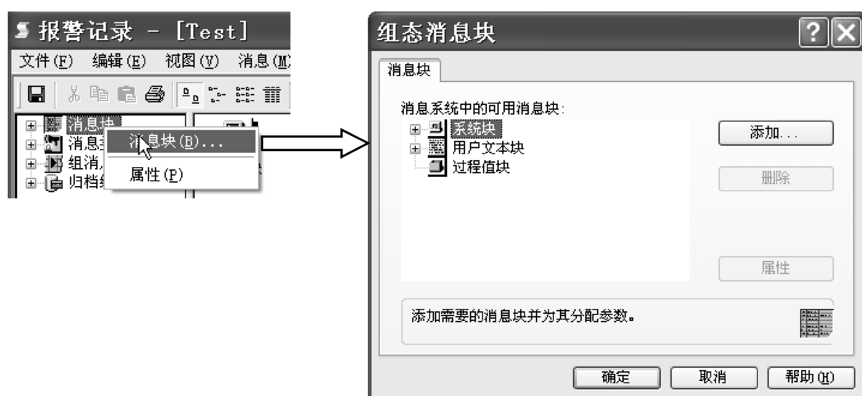



图 5-7 打开“组态消息块”对话框

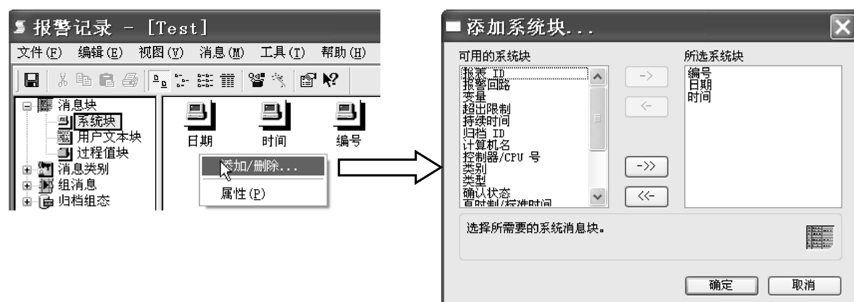



图 5-8 添加/删除消息块的方法

## 2. 消息块属性的修改

在图 5-7 所示的“消息系统中的可用消息块”框中选择一个消息块，例如系统块中的“日期”消息块，单击“属性”按钮，打开所选消息块的属性对话框。也可以在报警记录编辑器的浏览窗口选中消息块所属的类型，例如“系统块”，然后在数据窗口中用鼠标右键单击需要修改属性的消息块，例如“日期”消息块，在弹出的菜单中选择“属性”，打开该消息块的属性对话框，如图 5-9 所示。

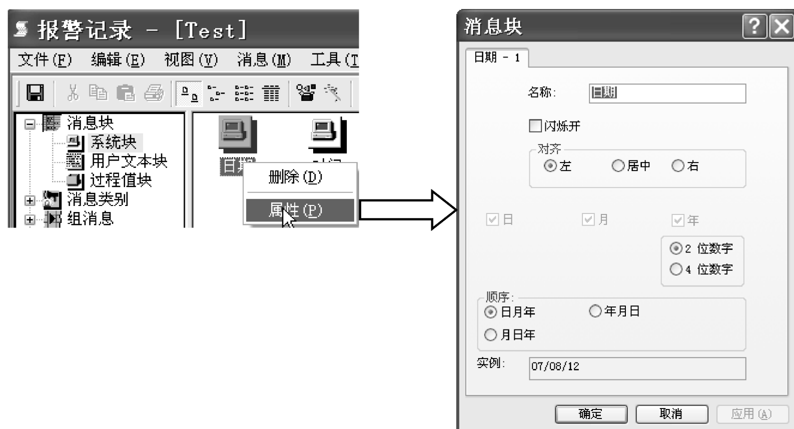



图 5-9 打开消息块的属性对话框



在消息块的属性对话框中，可以修改该消息块的参数，例如“日期”消息块的名称、对齐方式、日期格式等参数。不同的消息块，可供修改的参数也不相同。

### 5.2.5　组态消息类别和消息类型

组态消息系统时，必须为每条消息分配一个消息类别。这样可以不必单独为每个消息定义大量基本设置，而只需为整个消息类别定义全局应用的设置。

WinCC 提供 16 个消息类别和两个预设的系统消息类别。系统消息是在系统内部生成的消息，如来自操作员输入或系统故障的消息。WinCC 为处理系统消息提供特别的系统类别，不可添加新的系统类别。提供有下列标准消息类别：

- 错误。
- 系统，需要确认。
- 系统，没有确认。

组态消息类别时，主要进行以下基本设置：

- 消息类别的名称。
- 已分配的消息类型。
- 确认方法和相关的状态文本。
- 为一个消息类别的所有消息类型定义输出声音/光学信号。

**1. 消息类别的添加/删除**

在报警记录编辑器的浏览窗口中，用鼠标右键单击“消息类别”文件夹，在弹出的菜单中选择“添加/删除”，将打开“添加消息类别…”对话框，如图 5-10 所示。

在“可用的消息类别”列表中选择需要添加的消息类别。单击[->]按钮将该消息类别添加至“选定的消息类别”列表中。单击[->>]按钮，可将“可用的消息类别”列表中所有的消息类别添加至“选定的消息类别”列表中。

在“选定的消息类别”列表中选择需要删除的消息类别，单击[<-]将该消息类别移动至“可用的消息类别”列表。单击按钮[<<-]可删除“选定的消息类别”列表中所有的消息类别，但不能删除“系统，需要确认”和“系统，没有确认”消息类别。

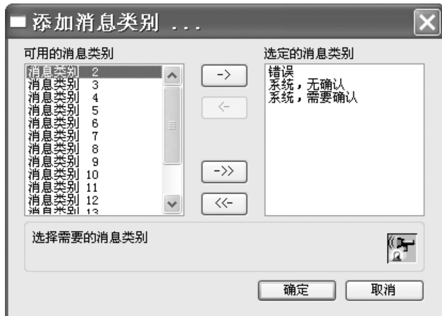



图 5-10　“添加消息类别…”对话框

**2. 编辑消息类别**

在报警记录编辑器的浏览窗口中选择“消息类别”文件夹，在数据窗口中用鼠标右键单击要编辑的消息类别，在快捷菜单中选择“属性”，或双击要编辑的消息类别，打开所选消息类别的组态对话框，如图 5-11 所示。

在“组态消息类别”对话框中，有三个选项卡可供选择。选择“类别 1”（编号会因消息类别不同而变化）选项卡，可以修改类别的名称、添加或删除该消息类别包含的消息类型以及编辑其包含的消息类型的属性；选择“确认”选项卡，可以设置消息类别的确认方式以及指定如何完成中心设备的确认；选择“状态文本”选项卡，可以设置在消

息行中“状态”和“确认状态”系统块所显示的文本。

### 3. 为消息类别添加/删除消息类型

在报警记录编辑器的浏览窗口中用鼠标右键单击要添加或删除消息类型的消息类别，在弹出的快捷菜单中选择“添加/删除消息类型...”，打开“添加消息类型...”对话框，如图5-12所示。




图5-11　消息类别的组态对话框

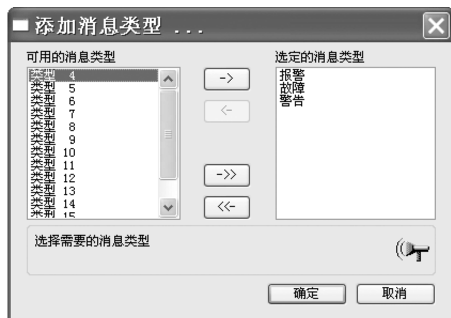



图5-12　“添加消息类型…”对话框

在“可用的消息类型”列表中选择需要添加的消息类型。单击 -> 按钮将所选的消息类型添加至“选定的消息类型”列表中。要添加所有可用的消息类型，单击 ->> 按钮。总共可以为一个消息类别添加16个消息类型。

在“选定的消息型”列表中选择需要删除的消息类型，单击 <- 按钮将该消息类型移动至“可用的消息类型”列表。单击按钮 <<- 可删除“选定的消息类型”列表中所有的消息类型。

还可以在打开图5-11所示的消息类别的组态对话框并选择第一个选项卡后，单击“添加…”按钮打开“添加消息类型…”对话框来添加或删除消息类型。

### 4. 更改消息类型的属性

可以在消息类型的属性对话框中更改消息类型的属性，包括修改消息类型的名称以及设置消息在“到达”（进入）、“离开”和“确认”状态时显示的颜色（文本颜色和背景颜色）。

在报警记录编辑器的浏览窗口中选择要更改消息类型属性的消息类别，在数据窗口中用鼠标右键单击要做更改的消息类型，在弹出的快捷菜单中选择“属性”命令，打开“类型”对话框，如图5-13所示。

还可以在打开图5-11所示的消息类别的组态对话框并选择第一个选项卡后，单击“属性”按钮，打开“类型”对话框。

### 5. 组态消息类别的确认

在打开消息类别的组态对话框并选择“确认”选项卡后，可以设置消息类别的确认方式以及指定如何完成中心设备的确认，如图5-14所示。






图 5-13 “类型”对话框




图 5-14 组态消息类别的确认

“确认理论”框中各选项的含义见表 5-1。

**表 5-1 确认理论选项的含义**

| 选　　项 | 描　　述 |
|---|---|
| 确认到达 | 选中带到达确认的单个消息的复选框。必须确认该消息类别的到达消息。否则，在确认前，消息一直保留未决状态 |
| 确认离开 | 选中带有双模式确认的单个消息的复选框。必须确认该消息类别的离开消息 |
| 闪烁开始 | 选中带单模式或双模式确认的新数值消息的复选框。当其显示在消息窗口中时，该消息类别的消息将闪烁。为了使消息的消息块在运行系统中闪烁，则必须在相关消息块的属性中启用闪烁 |
| 只是初始值 | 选中带单模式确认的初始值消息的复选框。只有此消息类别中的第一条消息在消息窗口中显示闪烁，“闪烁开”复选框必须预先选中 |
| 不带“离开”状态的消息 | 选中带或不带确认的无“离开”状态的消息的复选框。如果选择该选项，消息不具有“离开”状态。如果消息仅识别“已到达”状态，则不将该消息输入到消息窗口中，而是只进行归档 |
| 将注释指定给唯一用户 | 如果选中了此复选框，则将消息窗口中的注释分配给已登录的用户。在“用户名”系统块中输入用户。如果至今没有输入任何注释，则任何用户都可输入第一个注释。在输入第一个注释之后，所有其他用户只能对注释进行读访问 |
| 到达消息的注释始终在消息列表中 | 如果选中了此复选框，到达消息的注释总是与动态组件“@100%s@”、“@101%s@”、“@102%s@”和“@103%s@”一起显示在用户文本块中。随后的显示则取决于消息列表中消息的状态 |

“确认中央信令设备”框中各选项的含义见表 5-2。

**表 5-2 确认中央信令设备选项的含义**

| 选　　项 | 描　　述 |
|---|---|
| 分开确认键 | 消息到达时所触发的中央信令设备，将由用户通过工具栏中的单独确认按钮或键盘上的某个键进行确认。必须在消息窗口中组态确认按钮。即使在已经选择了“通过单独确认”选项时，也始终激活“分开确认键” |
| 通过单独确认 | 消息到达时被触发的中央信令设备是通过确认触发它们的消息来进行确认的 |
| 变量 | 中央信令设备由所定义的变量进行控制 |
| R | 打开“选择变量”对话框来选择一个变量作为确认变量 |

6．组态消息类别的状态文本

在打开消息类别的组态对话框并选择“状态文本”选项卡后，可以设置在消息行中“状态”和“确认状态”系统块所显示的文本，如图 5-15 所示。




图 5-15　组态消息类别的状态文本

消息类别的组态对话框的“状态文本”选项卡的各选项的含义见表 5-3。

表 5-3　“状态文本”选项卡各选项的含义

| 选　项 | 描　述 |
| --- | --- |
| 已到达 | 切换到待报告操作状态时的“到达的”消息文本 |
| 已离开 | 退出待报告操作状态时的“离开的”消息文本 |
| 已确认 | 确认消息的文本 |
| 已到达和离开 | 已到达并离开的消息文本 |

## 5.2.6　组态单个消息

组态单个消息用于给每个事件分配一条消息，单个消息由定义的消息块组成。多种单个消息可以总括为一个组消息。

可在报警记录编辑器的表格窗口中组态单个消息，主要包括创建和复制单个消息、删除单个消息以及定义和修改单个消息的参数。

1．创建、复制和删除单个消息

在报警记录编辑器的表格窗口中，可以通过插入新的一行或复制并粘贴已存在的单个消息来创建一条单个消息。

（1）创建一条新的单个消息

在表格窗口的任意位置单击鼠标右键，都可以弹出快捷菜单，在菜单中选择“添加新行”命令，将添加一行到表格末尾。该行包含了新消息的数据，如果需要，可以更改单个消息的属性。

（2）复制并粘贴一条单个消息

在表格窗口中，用鼠标右键单击要复制的单个消息，在弹出的快捷菜单中选择“复制行”命令，该单个消息将被复制到剪贴板。在表格窗口的任意位置单击鼠标右键，在



弹出的快捷菜单中选择“添加复制行”，将添加一行到表格末尾。该行包含已复制消息的数据，如果需要，可以更改已复制单个消息的属性。

（3）删除一条单个消息

在表格窗口中，用鼠标右键单击要删除的单个消息，在弹出的快捷菜单中选择“删除行”命令，该单个消息将从表格中删除。

**2. 单个消息的参数**

在单个消息的属性对话框——“单个消息”对话框的“参数”选项卡上可以指定下列设置：

- 单个消息参数，例如对消息类别和消息类型的分配。
- 用于连接各种不同变量的参数，例如消息变量。

在报警记录编辑器的表格窗口中用鼠标右键单击选定的单个消息，在弹出的快捷菜单中选择“属性”，可以打开“单个消息”对话框，如图 5-16 所示。在“单个消息”对话框的“参数”选项卡中可以设置的单个消息参数见表 5-4。

在对话框中，单击[...]按钮可以打开“变量－项目”对话框选择变量，单击[R]按钮可以复位变量分配以及其中所使用的位。




图 5-16　“单个消息”对话框

**表 5-4　单个消息的参数表**

| 参　　数 | 说　　明 |
| --- | --- |
| 编号 | 单个消息的编号。只能在表格窗口中设置该编号 |
| 类别 | 单个消息的消息类别 |
| 类型 | 单个消息的消息类型 |
| 组 | 将单个消息分配给用户自定义的组消息时，可从选择列表中选择一个已组态的组消息 |
| 隐藏掩码 | 显示要隐藏基于十六进制数值的消息的条件。在系统运行期间，如果变量编号的值与隐藏屏幕的位位置匹配，则消息会在消息列表、短期归档列表和长期归档列表中自动隐藏。必须将单个消息指派给自定义组消息，且对于组消息，隐藏变量必须可组态 |

（续）

| 参　数 | 说　明 |
|---|---|
| 优先级 | 定义消息的优先级。可根据优先级，对显示的消息进行选择和分类。数值范围是“0”～“16”。在 WinCC 中，没有指定对应于最高优先级的数值。在 PCS7 环境中，数值 16 对应于最高优先级 |
| 仅为单个确认 | 必须单独确认该消息。无法使用组确认按钮进行确认 |
| 控制中央信令设备 | 在消息到达时将触发中央信令设备 |
| 将被归档 | 消息将被保存在归档中 |
| 是在下降沿创建的 | 在位消息的处理过程中，可指定在信号的上升沿或下降沿创建消息。在所有其他消息的处理过程期间，始终在信号上升沿创建消息。对于在信号下降沿创建的消息，消息变量必须将起始值组态为“1” |
| 触发一个动作 | 该消息触发标准函数 GMsgFunction，使用“全局脚本”编辑器可对该函数进行编辑。使用全局脚本函数浏览器可定位“标准函数/报警”下的函数 |
| PLC 编号 | 触发了消息的 PLC 的编号。<br>只有在系统消息块中有可供使用的“AS/CPU 编号”块时，才可显示该域。<br>此处所组态的值在运行系统时显示在各单个消息的消息块中，并在与 AS 的通信中不发挥任何作用 |
| CPU 编号 | 触发了消息的 CPU 编号。<br>只有在系统消息块中有可供使用的“AS/CPU 编号”块时，才可显示该域。<br>这里所组态的值在运行系统时显示在各单个消息的消息块中，并在与 AS 或 CPU 的通信中不发挥任何作用 |
| 消息变量 | 消息变量包含用于触发当前选定消息的位 |
| 消息位 | 消息变量内的位编号，用来触发当前选定消息 |
| 确认变量 | 此处指定将要作为确认变量使用的变量 |
| 确认位 | 确认变量内用于确认消息的位编号 |
| 状态变量 | 此处指定将在其中存储单个消息状态的变量（“已到达/已离开”和确认状态） |
| 状态位 | 状态变量内指示消息状态的位编号。用于强制确认的位将自动确定 |
| 格式 DLL | 如果状态变量是原始数据变量，此处必须选择相关联的编译程序 |
| DLL 参数 | 此处输入指定接口的 DLL 消息参数（格式 DLL）。仅当该消息与通过 ODK 链接的单独“格式 DLL”相关时，才需进行该组态 |

### 3. 单个消息的变量

组态单个消息时，如果有必要，可以对单个消息的消息变量、状态变量和确认变量进行组态。在选定的单个消息的快捷菜单中，选择“属性”命令，打开“单个消息”对话框，在“参数”选项卡中可以指定单个消息的变量。单击[...]按钮可以打开“变量－项目”对话框选择变量，单击[R]按钮可以复位变量分配和消息位。

（1）单个消息的消息变量

消息通过消息变量触发，下列组态选项可用于消息变量：

- 为每条单个消息组态一个单独的消息变量。
- 通过一个消息变量触发多条单个消息。通过消息位区别单个消息。

（2）单个消息的状态变量

单个消息包含两个关键的消息状态：



- “已到达/已离开”状态表示消息是否到达或离开。
- 确认状态表示消息是否需要确认。

单个消息的这两种状态存储在状态变量中。根据变量的数据类型，最多可在一个状态变量中记录 16 条单个消息，每条单个消息在状态变量中占用 2 位，即确认位和“已到达/已离开”状态位。

一旦接收到需要确认的单个消息且尚未确认时，状态变量中的确认位更改为“1”。一旦确认了需要确认的单个消息，确认位则更改为“0”。

状态变量中“已到达/已离开”状态位的位置由状态位标记。确认位的位置取决于状态变量的数据类型。与包含有“已到达/已离开”状态位的距离是：

- “8 位无符号数”变量：4 位。
- “16 位无符号数”变量：8 位。
- “32 位无符号数”变量：16 位。

例如，如果状态变量为“32 位无符号数”类型，且状态位 =9，则：该状态变量的位号 9 表示单个消息的“已到达/已离开”状态。状态变量的位号 25 表示该单个消息是否需要确认。相应地，在“32 位无符号数”的状态变量中，位“0 ~ 15”分别对应于位“16 ~ 31”。

再如，如果状态变量为“16 位无符号数”类型，且状态位 =3，则：状态变量的位号 3 表示单个消息的“已到达/已离开”状态。状态变量的位号 11 表示该单个消息是否需要确认。相应地，在“16 位”的状态变量中，位“0 ~ 7”分别对应于位“8 ~ 15”。

（3）单个消息的确认变量

在单个消息中，一个确认变量的确认位用于触发确认和显示状态。适用规则如下：

- 如果单个消息的确认位数值为“1”，则表示已确认单个消息。
- 如果单个消息的确认位数值为“0”，则表示尚未确认单个消息。

在运行系统时确认一条单个消息时，设置该确认位。下列组态选项可用于确认变量：

- 为每条单个消息组态一个单独的确认变量。
- 在一个确认变量中组合多条单个消息，它们的区别通过确认位体现。

**4. 指定单个消息的消息文本**

可在单个消息的消息文本中指定该消息的一些信息，例如，触发该消息的有关故障原因或故障位置的信息。

在“单个消息”对话框中选择“文本”选项卡，可以指定单个消息的消息文本，如图 5-17 所示，“信息文本”框中可输入最多包含 255 个字符的信息，不能在系统运行时修改信息文本。块 1 ~ 10 代表相应的用户文本块，可为每个块输入最多包含 255 个字符的文本。

**5. 在用户文本块中插入过程值**

可在用户文本块中插入过程变量的值，但禁止超出用户文本块的最大长度（即 255 个字符）。在“单个消息”对话框中选择“文本”选项卡（参考图 5-17），单击需要插入过程变量的用户文本框后的□按钮，将打开“插入过程值”对话框，如图 5-18 所示。




图 5–17　指定单个消息的消息文本




图 5–18　“插入过程值”对话框

“插入过程值”对话框中的参数及其含义见表 5–5。

表 5–5　“插入过程值”对话框的参数及其含义

| 参　数 | 描　述 |
|---|---|
| 用户文本块 | 表示要插入过程值的用户文本块 |
| 长度 | 表示用户文本块中定义的文本长度 |
| 文本 | 镜像用户文本 |
| 过程值块 | 此处选择要插入的数值所对应的过程变量 |
| 类型 | 定义要插入的过程值的格式类型。<br>可用选项如下：文本；定点数（十进制、八进制、十六进制）；浮点数 |
| 位置 | 确定文本中格式声明的位置。如果指定“位置：10”，过程值将被插入到用户文本的第十个字符之后 |
| 数字的位数 | 定义在用户文本中为过程值保留的数字位数 |
| 精确度 | 定义要插入到用户文本中的过程值的数字位数 |
| 填充“0” | 如果激活该功能，则浮点数将在达到“精度”文本域中指定的数字位数后进行填充 |
| 左对齐 | 如果过程值中要插入的数字位数小于预留的数字位数，则使用该功能来指定在用户文本中插入过程值时是右对齐还是左对齐 |
| 格式说明 | 表示已指定的格式说明 |
| 实例 | 显示实例，用以说明与数字位数、精度、左对齐和填充相关的规范是如何影响不同类型的过程值的 |

## 5.2.7　组态模拟量报警

模拟量报警用于变量的限制值监控，可为变量指定任意多个限制值。如果超过了某个限制值，则在运行系统中生成一个消息。模拟量报警是一个 WinCC 的附加件，通过选择报警记录编辑器“工具”菜单的“附加项”命令，或单击工具栏的图标，打开“附加项”对话框，将模拟量报警集成到消息系统中，“模拟量报警”条目将在报警记录编辑器的浏览窗口中显示。



将模拟量报警集成到消息系统以后，组态模拟量报警的过程如下：

1）设置监控变量。

2）指定监控的限制值。

**1. 设置监控变量**

在报警记录编辑器浏览窗口用鼠标右键单击“模拟量报警”，在弹出的菜单中选择“新建”，将打开监控变量的“属性”对话框，如图 5-19 所示。在“要监视的变量”框中输入变量名或单击[...]按钮，在随后打开的“变量 - 项目”对话框中选择变量。

除了设置监控变量，监控变量的相关属性同样在其“属性”对话框中进行设置，“属性”对话框可以设置的监控变量属性见表 5-6。




图 5-19　监控变量的“属性”对话框

**表 5-6　监控变量可设置的属性**

| 参　　数 | 描　　述 |
|---|---|
| 限制值数量 | 此处显示要监视的限制值数量 |
| 一条消息对应所有限制值 | 如果选中此选项，则在运行系统时，当超出其中一个已组态的限制值时，将始终显示同一条消息 |
| 消息号 | 在此处指定在运行系统时，当超出限制值时要生成的消息编号 |
| 延迟时间 | 在此处指定限制值超出范围和生成消息之间的延迟时间。消息只有在限制值超出范围达到整个延迟时间后触发。延迟时间可以设置为 250 ~ 10 000 ms。如果不需要延迟时间，输入数值“0” |

**2. 指定监控的限制值**

可为监控变量指定任意多个要监控的限制值。为此，需要多次调用限制值对话框来指定限制值。

在报警记录编辑器浏览窗口的“模拟量报警”文件夹中或模拟量报警的数据窗口中选择要监控的变量。在快捷菜单中选择“新建 ...”命令，将打开限制值的“属性”对话框，如图 5-20 所示，可以设置的限制值属性见表 5-7。

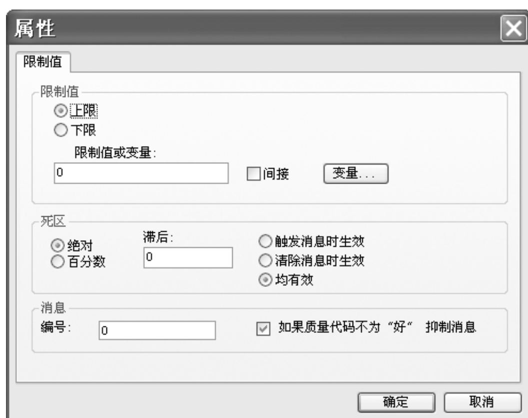



图5-20　限制值的“属性”对话框

**表5-7　限制值可设置的属性**

| 选　　项 | 描　　述 |
|---|---|
| 上限 | 当变量值超出上限值时生成消息 |
| 下限 | 当变量值低于下限值时生成消息 |
| 限制值或变量 | 编辑框中包含要监视的限制值或指定的变量。单击“变量”，打开变量选择对话框。<br>单击“确认”按钮时关闭变量选择对话框，将选中“间接”选项。不要使用其名称完全由数字组成的变量来定义间接限制值。变量名称必须以字母打头。通过间接限制值规范，在触发消息时，将要监控的变量值存储在过程值块3中 |
| 滞后 | 如果已定义了滞后，消息不会在达到其限制值时立即触发；而是在实际变量值超出或低于限制值后，按照滞后值触发。<br>滞后既可指定为绝对值，也可是偏离限制值的百分比。根据所选的设置，滞后将在生成或收回消息时生效 |
| 消息 | 在此处输入在运行系统期间超出限制值时生成的消息的编号。<br>只有在“变量的属性”对话框中没选择“一条消息对应所有限制值”时，才能修改该域 |
| 抑制消息 | 可使用该选项来抑制其质量代码不符合“好”状态的变量的消息，然后将不会检查变量的所有数值更改是否超出限制 |

注意：当组态模拟量报警时，禁止通过报警记录编辑器的表格窗口再次创建自动生成的消息。模拟量报警组态完成并保存后，需要退出报警记录编辑器，然后重新启动报警记录编辑器，组态的模拟量限制值消息在数据窗口中显示，且可在表格窗口中进行更改。

限制值的数值、滞后和触发数值均存储在过程值块1~3中。过程值块4~10不能用于模拟量限制值消息。

### 5.2.8　组态消息归档

WinCC的归档管理器提供了归档消息的功能，这样可以为操作和错误状态创建指定的文档。从WinCC V6.0开始，Microsoft SQL Server用于归档。例如，WinCC V6.2使用SQL Server 2005作为归档的服务器。



每当相关事件产生时，例如出现错误或超出限制值，将在运行系统中输出报警记录编辑器中所组态的消息。如果如下消息事件产生，则系统对消息进行归档：

- 当消息产生时。
- 当消息的状态改变时（例如从“消息到达”到“确认消息”）

可在归档数据库中保存消息事件，并将其归档为消息报表。在数据库中归档的消息可在运行系统中输出，例如在消息窗口中输出。

与消息有关的所有数据（包括组态数据）将保存在消息归档中。因此可从归档读出一条消息的所有属性，包括消息类型、时间标志和文本等。消息的组态数据发生变化时，将用新组态的数据创建新归档，而对变化前归档的消息不产生影响。表 5-8 列出了 WinCC 系统可以处理的消息数目，实际数字取决于所使用的服务器，表 5-8 中的信息基于以下服务器配置：

- Intel Pentium Ⅲ 1.4 GHz。
- 512 MB RAM。
- 40 GB 硬盘。
- 最多 16 台客户机。

**表 5-8　WinCC 系统可处理的消息数目**

| | 服务器 | 特殊归档服务器 |
|---|---|---|
| 连续信息装载量（每秒） | 10 条消息 | 100 条消息 |
| 消息流量（每 10 秒） | 2 000 条消息 | 15 000 条消息 |

**1. 组态要归档的消息**

可从所有已组态的消息中选择要归档的消息。实际上，WinCC 默认归档新建的消息。因此，从原则上说，只需要按下面所述的方法和步骤进行操作，为已修改了默认归档状态的消息重新激活归档：

1）打开报警记录编辑器。

2）在报警记录编辑器的表格窗口中用鼠标右键单击要归档的消息，在弹出的菜单中选择“属性”命令，打开该消息的“属性”对话框。

3）在消息的“属性”对话框中选择“参数”选项卡，并选中“该消息将被归档”复选框，如图 5-21 所示。

4）单击“确定”按钮保存所进行的设置，并退出“属性”对话框。

**2. 组态消息归档的方法和步骤**

可在报警记录编辑器中组态消息归档。组态时必须确保归档大小不超出可用的空闲内存空间。归档管理器将检查所选设置的真实性。在启动和结束 WinCC 运行系统时，大量已链接的数据库分段会导致系统出现较长的等待周期。

组态消息归档的方法和步骤如下：

1）打开报警记录编辑器，并在浏览窗口中选择“归档组态”。

2）用鼠标右键单击“消息归档”，在弹出的菜单中选择“属性”命令，将打开“AlarmLogging”对话框，如图 5-22 所示。

3）在“AlarmLogging”对话框中选择“归档组态”选项卡，为归档作下列设置：








图 5-21　消息归档的激活




图 5-22　“AlarmLogging”对话框

- 所有分段的时间段以及最大尺寸。这个规范定义了归档数据库的大小。如果超出其中一个标准，则启动新的分段并删除最旧的分段。
- 单个分段所包含的时间段和单个分段最大尺寸。如果违反了其中任意一个限制，则将启动一个新的单个分段。如果超出“所有分段的时间段”标准的限制，则最早的单个分段也将被删除。

4）在“更改分段的时间”域中输入首次更改分段的开始日期和开始时间。开始新的单个分段时，也要考虑组态“更改分段的时间”。

5）单击“确定”按钮保存所作的设置，并退出对话框。

在图 5-22 中，分段的首次改变时间是 5 月 1 日的 23：59。下一个与时间有关的分段的改变发生在“单个分段所包含的时间段”定义的周期中所组态的时刻。如果周期设置为“1 日”，则在每天的 23：59 更改分段。如果周期设置为“1 个月”，则在下个月的 1 日



的 23：59 发生分段改变。即使在启动运行系统后超出所组态的尺寸（“所有分段的最大尺寸”），分段也会改变。在这种情况下，最旧的单个分段也会被删除。

**3. 组态归档备份**

可以通过创建归档数据的定期备份，确保过程文档没有间隔。

（1）组态归档备份的方法和步骤

组态归档备份的方法和步骤如下：

1）打开报警记录编辑器，并在浏览窗口中选择“归档组态”。

2）用鼠标右键单击“消息归档”，在弹出的菜单中选择“属性”命令，打开“AlarmLogging”对话框，选择“备份组态”选项卡，如图 5-23 所示。




图 5-23　组态归档备份

3）如果已交换的归档数据要被签名，那么选中“激活签名”复选框。与 WinCC 重新连接时，通过签名可使得系统能确定在交换后是否修改了归档备份文件。

4）如果要备份归档数据，则选中“激活备份”复选框。此外，如果要在“目标路径”和“备选目标路径”两个目录下均保存已归档的数据，那么选中“备份到两个路径”复选框。

5）输入存储备份文件的目标路径。网络路径也可作为目标路径。在下列情况下使用“备选目标路径”：

- 备份介质上的存储空间已被占用。
- 原始目标路径不存在，例如，由于网络出现故障。

如果相关的系统消息已组态，则当指定的路径不可用时将输出消息。

6）单击“确定”按钮保存所作的设置，并退出对话框。

归档备份包括两个文件，其扩展名为 LDF 和 MDF。为传送归档备份（例如给另一台计算机），可复制相应的 LDF 和 MDF 文件。文件名的形式在本章 5.1.3 小节进行了详细介绍，在此不再赘述。

（2）归档备份的链接

为了在运行系统中访问归档备份的数据，需要将备份的相关数据库文件与对应的 WinCC 项目链接。可以在报警记录编辑器中手动建立项目与归档备份的链接，也可让系统自动建立链接。建立链接后，在运行系统的消息窗口中将显示已归档的消息。

在进行链接前，归档备份的相应的 LDF 文件和 MDF 文件位于组态计算机的本地目录

下，项目装载到组态计算机上，并处于运行状态。

在报警记录编辑器中手动建立项目与归档备份的链接的方法和步骤如下：

1）打开报警记录编辑器。

2）在报警记录浏览窗口中选择“归档组态”。在数据窗口中，用鼠标右键单击“消息归档”，在弹出的菜单中选择“链接归档”。

3）在打开的对话框中选择所需的数据库文件，然后单击“打开”。打开数据库该文件后，消息归档备份就链接到项目上了。

将归档备份文件添加到项目的“ProjectName\CommonArchiving”目录下，在运行系统中，消息归档将自动链接到项目。

（3）断开归档备份

如果在运行系统中不再需要访问归档备份中的数据，那么可从项目断开相关的数据库文件，方法如下：

1）打开报警记录编辑器。

2）在浏览窗口中选择“归档组态”，在数据窗口中使用鼠标右键单击“消息归档”，在弹出的菜单中选择“断开与归档的连接”。

3）选择所需的归档文件并单击“确定”按钮，断开归档文件链接。在运行系统中不能再访问备份归档的消息。

**4. 归档的清除**

当 WinCC 处于组态模式时，在报警记录编辑器的浏览窗口中选择“归档组态”，在数据窗口中使用鼠标右键单击“消息归档”，在弹出的菜单中选择“重置”，会将项目的“ProjectName\ArchiveManager\AlarmLogging”文件夹下的所有分段文件清除。当项目重新运行时，系统会自动重新创建相关的数据库文件。

## 5.3　在运行系统中显示消息

前文已述及，消息系统的运行系统负责完成报警消息的显示、确认和管理等，这主要通过 WinCC 报警控件（WinCC Alarm Control）实现。通过使用报警控件，用户在组态时可获得高度的灵活性，因为希望显示的消息窗口、消息行和消息块均可在图形编辑器中进行组态。在 WinCC 运行系统中，消息通过报警控件以表格的形式显示在过程画面中。

### 5.3.1　组态 WinCC 报警控件

假定当前项目中需要为 WinCC 报警控件组态了以下设置：

- 工具栏中的键盘功能。
- 消息窗口中的选择。
- 选择消息窗口中需要显示• 的消息块以及显示顺序。

组态步骤如下：

1）启动图形编辑器，打开新的画面。

2）将对象选项板中的“控件”→“WinCC Alarm Control”插入到画面中，并将其拖



到所期望的大小。

3）双击该控件，打开“WinCC 报警控件属性”对话框，如图 5-24 所示。选择“常规”选项卡，在“窗口标题”框中输入窗口名称，并选中“显示窗口标题”复选框，以便在运行系统中显示消息窗口的名称。

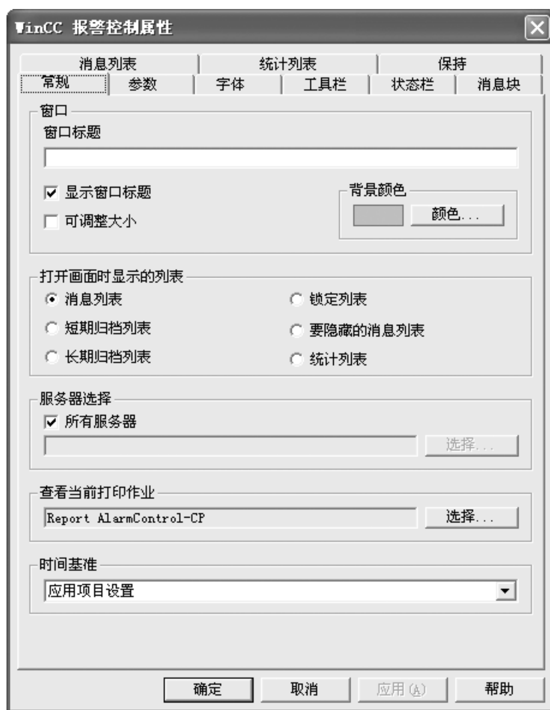



图 5-24 “WinCC 报警控件属性”对话框

4）选中“工具栏”选项卡中的“显示工具栏”复选框。在“键盘功能”下拉列表框中选择所需要的键盘功能，例如，需要以下键盘功能：“消息列表”、“锁定列表”、“选择对话框”、“锁定对话框”、“5 紧急确认”、“锁定/解锁消息”和“排序对话框”。

5）在“参数”选项卡的“选择”框中，选择“行”选项。

6）在“消息块”选项卡中，选择“时间”系统块以及“消息文本”和“错误点”两个用户文本块。

7）在“消息列表”选项卡中，指定要在消息窗口中显示的消息块以及它们的显示顺序。使用→按钮，将选定的消息块从“现有消息块”列表中移动到“选定消息块”列表，在“选定消息块”列表中列出的消息块将在运行系统的消息窗口中显示。通过选择一个消息块，然后单击或按钮更改消息块在消息窗口中显示的顺序。

8）单击“确定”按钮关闭对话框，并将设置保存在图形编辑器中。

### 5.3.2 报警控件在运行系统中的操作

在运行系统中，可使用工具栏按钮操作消息窗口。如果不希望使用工具栏来操作，则可使用“全局脚本”编辑器的“标准函数/报警”组中的可用函数进行操作。消息窗口部分工具栏按钮含义见表 5-9。

**表5-9 消息窗口工具栏部分按钮含义**

| 图标 | 名称 | 含义 |
|---|---|---|
| | 消息列表 | 显示当前未决的消息 |
| | 短期归档列表 | 显示短期归档列表中的已归档消息 |
| | 长期归档列表 | 显示长期归档列表中的已归档消息 |
| | 锁定列表 | 显示系统中所有已锁定的消息 |
| | 统计列表 | 显示在报警控件的“统计列表”标签上所组态的消息块和统计信息 |
| | 确认中央信号设备 | 确认视觉或听觉传感器 |
| | 单个确认 | 确认所选择的单个消息 |
| | 组确认 | 确认消息窗口中需要进行确认的所有未决的和可见的消息，除非这些消息需要单个确认 |
| | 自动滚动 | 如果激活了“自动滚动”，则将选择消息窗口中的最新消息。如果需要，可以移动消息窗口的可见区域。如果“自动滚动”未激活，则出现的新消息将不会处于选中状态。消息窗口的可见区域不变。仅在“自动滚动”尚未激活时，才可以有目标的选择消息行 |
| | 选择对话框 | 指定要在消息窗口中显示的消息的选择标准。符合这些标准的消息将不显示，但仍进行归档 |
| | “显示选项”对话框 | 定义哪些消息要显示在消息窗口中。<br>如果“所有消息”选项已激活，则消息窗口将同时显示隐藏的消息和显示的消息。<br>如果“显示的消息”选项已激活，则消息窗口仅显示显示的消息。<br>如果“隐藏的消息”选项已激活，则消息窗口仅显示隐藏的消息 |
| | “块”对话框 | 定义成块标准。满足这些标准的所有消息均不显示，也不进行归档 |
| | 打印信息报表 | 生成包含用户自定义的消息选择的文档的功能 |
| | 打印当前视图 | 开始打印输出在消息窗口中显示的消息。可在“常规”选项卡的控件属性中指定打印期间所使用的打印作业。在与打印顺序链接的布局的“参数”对话框中，选择要打印的消息列表 |
| | 紧急确认 | 紧急确认需要确认的消息。使用此功能，所选择的单个消息的确认位将直接发送到AS，即使消息并不是未决的 |
| | 显示第一条消息 | 选择第一条当前未决消息。如果需要，可以移动消息窗口的可见区域。只有在取消激活“自动滚动”时，此按钮才可用 |
| | 显示最后一条消息 | 选择最后一条当前未决消息。如果需要，可以移动消息窗口的可见区域。只有在取消激活“自动滚动”时，此按钮才可用 |
| | 显示下一条消息 | 选择下一条消息（相对于当前所选择的消息）。如果需要，可以移动消息窗口的可见区域。只有在取消激活“自动滚动”时，此按钮才可用 |
| | 显示前一条消息 | 选择前一条消息（相对于当前所选择的消息）。如果需要，可以移动消息窗口的可见区域。只有在取消激活“自动滚动”时，此按钮才可用 |
| | 信息文本对话框 | 打开对话框，显示信息文本 |
| | 注释对话框 | 打开文本编辑器，输入注释。此按钮只有在长期归档窗口中才可用。因此，该功能的组态仅对此窗口类型的“报警控件”属性有效 |
| | 报警回路 | 显示与所选消息相关的画面或触发脚本。也可以通过双击消息触发此功能 |
| | 解锁消息 | 单击该按钮解锁锁定列表中选择的消息 |



（续）

| 图　标 | 名　称 | 含　义 |
|---|---|---|
| | 锁定消息 | 单击此按钮锁定当前消息列表和消息归档列表中选择的消息 |
| | 排序对话框 | 打开对话框，设置所显示消息的用户自定义排序标准。用户自定义排序标准优先于“MsgCtrlFlags”属性生成的排序标准 |
| | 时间基准对话框 | 打开对话框，为消息中显示的时间数据设置时间基准 |
| | 已隐藏消息列表 | 已隐藏消息列表中显示由于自动或手动隐藏而无法在消息列表、短期归档列表或长期归档列表中看见的所有消息 |
| | 隐藏消息 | 隐藏已在消息列表、短期归档列表或长期归档列表中选择的消息。消息在“已隐藏消息列表”中输入 |
| | 显示消息 | 再次显示在消息列表、短期归档列表或长期归档列表的“已隐藏消息列表”中所选择的消息。该消息将从“已隐藏消息列表”中删除 |

### 5.3.3　在运行系统中显示归档的消息

在运行系统中使用 WinCC 报警控件，不仅可显示当前消息，还可从归档调出消息。可在短期或长期归档列表中显示归档消息，短期归档列表也包含当前消息。

所有要在运行系统中显示的归档数据必须保存在本地归档服务器中，SQL Server 不允许对保存在别处的备份文件作任何访问，比如网络中的另一台计算机。

下列过程描述了在运行系统中显示归档消息的组态步骤：

1）在图形编辑器中，将 WinCC Alarm Control 添加到画面中。

2）在弹出的组态对话框中，选中“显示工具栏”复选框，并单击“确定”按钮确认。

3）双击该控件，打开“WinCC 报警控制属性”对话框。

4）选择“常规”选项卡，在“打开画面时显示的列表”框中，可以选择调用画面时报警控件默认要显示的消息，如图 5-26 所示。

- 消息列表：显示来自过程的当前消息。
- 短期归档列表/长期归档列表：当调用画面时，显示来自消息归档的消息。按各自的按钮，可以在显示当前消息和显示归档消息之间切换。
- 锁定列表：显示一系列被操作员锁定的消息。
- 统计列表：显示消息的统计信息。

5）为了在分布式系统中仅显示来自指定某些服务器的消息，可在“常规”选项卡的“服务器选择”框中，取消选中“所有服务器”复选框，然后单击“选择”按钮来选择服务器，如图 5-25 所示。

6）单击“工具栏”选项卡，在“键盘功能”下拉列表中至少选中“短期归档列表”或“长期归档列表”中的一个复选框。

7）单击“确定”按钮保存所作的设置。

在显示短期归档列表中的消息时，新到达的消息被立即更新。而显示长期归档列表中的消息时，情况与此不同。在短期归档列表中，不能输入或查看注释，只有在长期归档列表中才可能这样。




图 5-25　组态消息归档的显示

在运行系统中，操作员可以单击消息窗口工具栏的“短期归档列表”按钮或“长期归档列表”按钮在消息列表中显示来自归档的消息。操作员也可以采用与过滤当前消息相同的方式过滤来自归档的消息，例如只显示“故障”类型的消息或查看特定设备部分的消息。

## 5.4　应用实例

以组态鼓风炉（Blast furnace）的消息系统为例进行介绍。

### 5.4.1　组态要求

（1）消息系统的消息块

表 5-10 为鼓风炉的消息系统要求组态的消息块以及各消息块的属性

**表 5-10　鼓风炉消息系统的消息块**

| 消息块类型 | 消息块名称 | 消息块属性 |
|---|---|---|
| 系统块 | 日期 | 顺序：年月日，4 位年份 |
| | 时间 | 时间格式：小时分钟秒，12 小时制 |
| | 状态 | 长度：8 位字符 |
| | 编号 | 数位：6 位，以 0 开头 |
| | 类别 | 长度：8 位字符 |
| | 类型 | 长度：15 位字符 |



（续）

| 消息块类型 | 消息块名称 | 消息块属性 |
|---|---|---|
| 用户文本块 | 错误点 | 长度：10 位字符 |
| | 原因 | 长度：14 位字符 |
| | 消息文本 | 长度：12 位字符 |
| 过程值块 | 过程值：1 | 长度：4 位字符 |

（2）消息系统的消息类别和消息类型

表 5-11 为鼓风炉的消息系统要求组态的消息类别、消息类型以及对应的属性。

**表 5-11　鼓风炉消息系统的消息类别和消息类型**

| | 属性名称 | 属性值 |
|---|---|---|
| 消息类别 | 名称 | 鼓风炉 |
| | 消息类型 | 报警、失败 |
| | 确认 | 确认到达 |
| | 状态文字 | 到达：－ARR，离开：－DEP，已确认：－ACK，已到达和离开：－A/D。 |
| 消息类型 | 消息类型的名称 | 报警、失败 |
| | 文本颜色 | 到达：红色，离开：绿色，确认：黄色 |

（3）归档要求

鼓风炉的消息系统要求对消息进行归档，归档的要求如下：

- 单个分段的时间段：1 小时，单个分段的最大尺寸：100 MB。
- 所有分段的时间段：1 天，所有分段的最大尺寸：1000 MB。

（4）单个消息

要求在报警记录编辑器中生成 4 个单消息，详细要求见表 5-12。

**表 5-12　鼓风炉消息系统的单个消息组态要求**

| 编号 | 消息类别 | 消息类型 | 消息变量 | 消息变量位 | 消息文本 | 错误点 | 原　　因 |
|---|---|---|---|---|---|---|---|
| 1 | 鼓风炉 | 失败 | event1 | 0 | 转子 1 | 电动机 | 润滑不够 |
| 2 | 鼓风炉 | 报警 | event1 | 1 | 转数 | 电动机 | 转数太高 |
| 3 | 鼓风炉 | 报警 | event1 | 2 | 标称电流 | 电动机 | 超标称电流 |
| 4 | 鼓风炉 | 失败 | event1 | 3 | 过载 | 电动机 | 2 级过载 |

（5）WinCC 报警控件组态要求

鼓风炉消息系统的消息窗口（即 WinCC 报警控件）的组态要求如下：

- 常规：窗口标题——鼓风炉－W，窗口类型——Message Window。
- 消息行包含的消息块：日期、时间、编号、状态、消息文本、错误点、原因、类别和类型。
- 参数：分割线——水平和垂直的“激活”；选择——单元格“激活”；其他——自动滚动未被激活（窗口类型：消息窗口）；行属性——行标题“激活”；列属性

——列标题“激活”，改变列宽度“激活”。
- 状态栏：显示，向下对齐，选择状态栏全部元素。
- 工具栏：显示，向上对齐，激活全部键盘功能。

## 5.4.2　组态方法与步骤

根据上述组态要求，组态鼓风机的消息系统，详细步骤如下：

1）在 WinCC 项目管理器左侧浏览窗口中双击“报警记录”或鼠标右键单击“报警记录”，在弹出的菜单中选择“打开”，进入报警记录编辑器。

2）组态消息块：
- 在报警记录编辑器的浏览窗口中打开“消息块”的 + 号，打开消息块的对象结构，用鼠标右键单击“系统块”，在弹出的菜单中选择“添加/删除”命令，弹出“添加系统块...”对话框，根据表 5-10 从“可用系统块”下拉列表框中将日期、时间、编号、状态、类别和类型等系统块添加到“所选系统块”下拉列表框，单击“确定”按钮退出对话框。用同样的方法根据表 5-10 添加系统要求的用户文本块和过程值块。
- 根据表 5-10 的要求修改所添加的消息块的属性，以“日期”块为例。在报警记录编辑器的浏览窗口选中“系统块”，在数据窗口中用鼠标右键单击“日期”块，在弹出的菜单中选择“属性”命令，打开“消息块”对话框，如图 5-26 所示，修改“日期”消息块的属性后，单击“确定”按钮退出对话框。按同样的方法修改其他消息块的属性。




图 5-26　修改“日期”消息块的属性

3）组态消息类别和消息类型：
- 在报警记录编辑器浏览窗口用鼠标右键单击“消息类别”，在弹出的菜单中选择“添加/删除”命令，打开“添加消息类别…”对话框，从“可用的消息类别”列



表框中添加 1 个消息类别例如“消息类别 2”到“选定的消息类别”列表框，单击“确定”按钮，退出对话框。

- 在数据窗口用鼠标右键单击添加的消息类别，在弹出的菜单中选择“属性”命令，打开“组态消息类别”对话框，如图 5-27 所示。选择“类别 2”选项卡，在“类别名称”框中输入“鼓风炉”。单击“添加”按钮，打开“添加消息类型…”对话框，添加 2 个消息类型，例如“类型 17”和“类型 18”。
- 在“消息类型”列表中选中“类型 17”，单击“属性”按钮，打开“类型”对话框。在“消息类型的名称”框中输入类型名称“报警”。在“预览”框中选中“进入”，单击“文本颜色”按钮打开“颜色”对话框并设置“进入”状态时的文本颜色为“红色”，用同样的方法根据表 5-10 设置“离开”和“已确认”两种状态时的文本颜色。用同样的方法修改“类型 18”的属性。
- 在“组态消息类别”对话框中选择“确认”选项卡，在“确认理论”区域中选中“确认到达”复选框。
- 在“组态消息类别”对话框中选择“状态文本”选项卡，在“已到达”、“离开”、“已确认”和“已到达和离开”后的文本框中分别输入“ - ARR”、“ - DEP”、“ - ACK”和“ - A/D”。
- 在“组态消息类别”对话框中，单击“确定”按钮，保存所有设置并退出对话框。

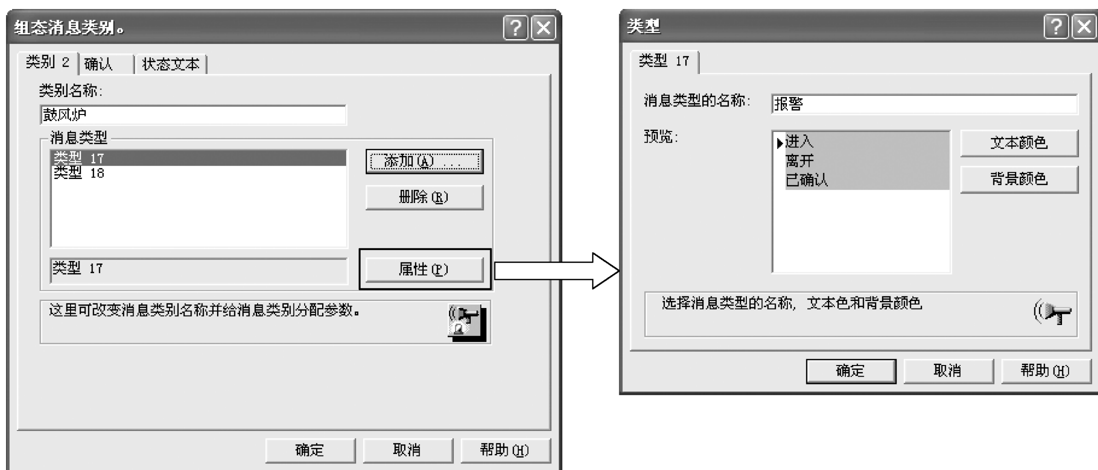



图 5-27　组态消息类别和消息类型

4）组态消息归档：

- 在报警记录编辑器浏览窗口单击“归档组态”，用鼠标右键单击“消息归档”，在弹出的菜单中选择“属性”，打开“AlarmLogging”对话框，如图 5-28 所示。
- 根据鼓风炉消息系统的归档要求设置单个分段的时间段为 1 小时，单个分段最大大小为 100 MB，所有分段的时间段为 1 天，所有分段的最大尺寸为 1000 MB。
- 单击“确定”按钮，保存所有设置并退出对话框。




图 5-28　组态归档

5）组态单个消息：

- 在 WinCC 变量管理器中，创建 1 个类型为“无符号 8 位数”的内部变量 event1 作为消息变量，创建 1 个类型为“无符号 16 位数”的内部变量 ProcessTag1 作为监控的过程变量。
- 在报警记录编辑器的表格窗口用鼠标右键单击，在弹出的菜单中选择“添加新行”命令，在表格窗口将添加 1 条新的单个消息。用鼠标右键单击新添加的行，在弹出的菜单中选择“属性”命令，打开“单个消息”对话框。
- 在“单个消息”对话框中选择“参数”选项卡，将“数”（编号）为 1 的单个消息的“类别”设置为“鼓风炉”，“类型”设置为“失败”。在“连接”框内，单击“消息变量”后的按钮[...]，打开“变量 – 项目”对话框选择变量 event1 为消息变量，设置消息位为 0，如图 5-29 左图所示。




图 5-29　组态单个消息



- 在“单个消息”对话框中选择“文本”选项卡，在“消息文本”后的文本框中输入“转子 1”，“错误点”后的文本框中输入“电动机”，在“原因”好的文本框中输入“润滑不够”，如图 5-29 右图所示。
- 在“单个消息”对话框中选择“变量/动作”选项卡，单击“过程值：1”后的按钮[...]，打开“变量 - 项目”对话框并选择变量 ProcessTag1。
- 用同样的方法根据表 5-10 组态编号 2 ~4 的单个消息。

6）组态 WinCC 报警控件：

- 打开 WinCC 图形编辑器，创建新的过程画面，将对象选项板中的“控件”→“WinCC Alarm Control”插入到画面中，并将其拖到所期望的大小。
- 双击该控件，打开“WinCC 报警控件属性”对话框，选择“常规”选项卡，在“窗口标题”框中输入“鼓风炉 - W”，并选中“显示窗口标题”的复选框。在“打开画面时显示的列表”框中选择“消息列表”。
- 选择“消息列表”选项卡，将“现有消息块”列表框中所有的消息块添加到“选定消息块”列表框中。
- 选择“参数”选项卡，在“分割线”框内选中“水平”和“垂直”的复选框。在“选择”框内激活“单元格”，“行属性”框内选中“行标题”复选框，“列属性”框内选中“列标题”的复选框和“改变列宽度”的复选框。
- 选择“状态栏”选项卡，选中“显示状态栏”和“状态栏的全部元素”复选框。
- 选择“工具栏”选项卡，选中“显示工具栏”和“全部键盘功能”的复选框。
- 单击“WinCC 报警控件属性”对话框的“确定”按钮，保存所有的设置并退出对话框。保存画面为“报警画面 . Pdl”。

7）为了在项目运行时测试消息系统的组态结果，在过程画面“报警画面 . Pdl”中添加 1 个对象选项板中的“标准”→“窗口对象”→“复选框”。

- 打开复选框的“对象属性”对话框，将属性“几何”→“框数量”的静态值设置为 4。
- 属性“字体”→“索引”为 1 时对应的“字体”→“文本”设置为“1 号消息”，属性“字体”→“索引”为 2 时对应的“字体”→“文本”设置为“2 号消息”，依次类推，“索引”为 3 时对应“3 号消息”，“索引”为 4 时对应“4 号消息”。
- 为属性“输出/输入”→“选择框”的动态列组态与变量 event1 的变量连接，并将“更新周期”设置为“有变化时”。
- 在图形编辑器中保存所作的组态。

8）激活报警记录的运行系统：在 WinCC 项目管理器的浏览窗口选择“计算机”，在数据窗口用鼠标右键单击当前组态消息系统的计算机名称，在弹出的菜单中选择“属性”命令，打开“计算机属性”对话框。选择“启动”选项卡，在“WinCC 运行时的启动顺序”列表框中选中“报警记录运行系统”的复选框，如图 5-30 所示。单击“确定”按钮，退出“计算机属性”对话框。

9）运行 WinCC 项目，并切换到“报警画面 . Pdl”，操作画面上的复选框测试消息系统，效果如图 5-31 所示。




图 5-30　激活报警记录运行系统




图 5-31　鼓风炉消息系统测试结果

## 5.5　组态技巧与技术问答

本节将介绍一些 WinCC 组态消息系统的技巧，并针对组态过程中遇到的一些常见的技术问题进行分析和解答。

### 5.5.1　消息窗口无法显示消息的解决方法

#### 1. 问题现象

如图 5-32 所示，在 WinCC 项目运行后，消息窗口无法显示消息，在消息窗口的中央显示报错信息“没有到数据服务器的连接!”，在消息窗口的状态栏显示“所有服务器的连接失败。”

#### 2. 问题原因

出现图 5-32 所示的问题最根本的原因是没有启动报警记录运行系统。






图 5-32 消息窗口无法显示消息

3. **解决方法**

在 WinCC 项目管理器的浏览窗口选择“计算机”，在数据窗口用鼠标右键单击当前组态消息系统的计算机名称，在弹出的菜单中选择“属性”命令，打开“计算机属性”对话框。选择“启动”选项卡，在“WinCC 运行时的启动顺序”列表框中选中“报警记录运行系统”的复选框。

另外需要注意的是，在报警记录编辑器中对消息系统做任何更改都需要保存后才能生效，否则无法在消息窗口中显示所作的更改。

## 5.5.2 WinCC 实现语音报警的组态方法

在工业现场安全是极其重要的，无论从哪个角度讲都应该利用一切手段减少故障的发生。在故障已经发生的时候，应该在第一时间以多种方式通知现场人员有故障发生。现代计算机的处理速度完全可以胜任对于图形、语音的同时处理。因此，可以利用计算机的声卡和音箱在有故障发生的时候产生语音报警，但是，WinCC 本身并不具备产生语音的功能，而且其内部也没有提供处理声音的函数。解决这个问题的方法归功于微软开发的 Windows 操作系统与 WinCC 地完美结合，因为在 WinCC 中可以直接调用 Windows 的 API 函数。实现的具体 C 脚本代码如下：

```
{
    #pragma code("Winmm.dll")
        void WINAPI PlaySoundA(char *pszSound, char *hmode, DWORD dwflay);
    #pragma code()
    if (GetTagBit("sound"))
    PlaySoundA("G:\WINDOWS\Media\Windows XP 错误.wav",NULL ,8);
    return 0;
}
```

在上述 C 脚本代码中，二进制变量 sound 定义故障是否出现（1：故障出现，0：故障消除)，因此，当变量 sound 值为 1 时，将触发语音报警。语音报警播放的语音文件是 Windows XP 操作系统的语音文件“Windows XP 错误.wav”，用户可以根据自己项目的需要选择播放其他语音文件。

有一个值得注意的问题是上述 C 代码的加入点。通过反复多次的实践得出结论，这

个点必须加在始终显示于屏幕上的任何图形元素的属性中，即需要针对该图形元素的属性组态 C 动作，且该 C 动作应该周期性的触发（标准周期或画面周期）。也可以将上述 C 代码组态到一个周期性触发的全局脚本动作中。这样才能达到语音报警的预期效果。

### 5.5.3　WinCC 组态来自于 PLC 的编号消息的方法

由本章 5.1.1 节的图 5-1 可知，由 WinCC 数据管理器的消息变量触发的消息包括位（布尔值）变量、模拟量（上下限报警）和消息帧。消息帧是 WinCC 使用一个 RawData（原始数据类型）变量获取 PLC 主动发送的所有消息帧，然后用特定格式的动态链接库从该数据帧中提取出相关的报警信息，在 HMI 上实现归档显示等功能。

消息帧方式一般常见于 WinCC（或 WinCC flexible）集成在 SIMATIC Manager 中的项目，该项目在 SIMATIC Manager 中经过简单设置和 OS 编译后，WinCC 项目中会自动生成与 PLC 的连接，并自动生成在 SIMATIC Manager 中创建的变量，其中就包括类型为 Raw Data 的消息帧变量，例如图 5-33 所示的变量 S7 $ 程序（1）#RawEvent。

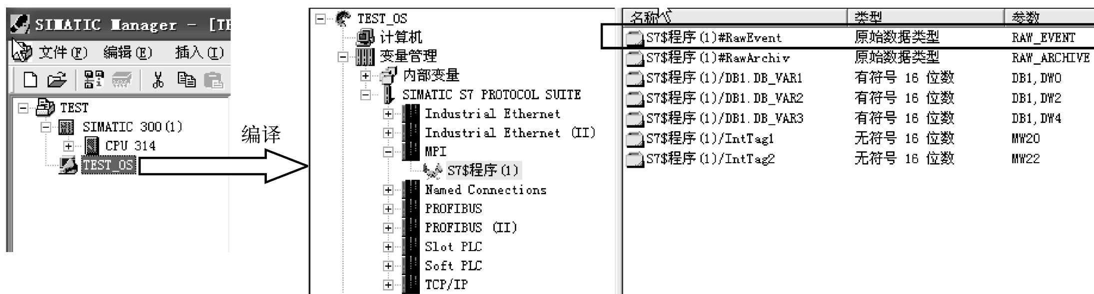



图 5-33　消息帧变量的生成

消息帧的消息实现方式类似于 PCS 7 的消息实现方式，消息组态在 STEP 7 中完成，消息经编译进入 WinCC 报警记录中，并且每条消息被自动分配了不同的消息号，这种消息被称为编号消息，它又可以分为块消息和符号表消息。

块消息是指 PLC 通过调用相应的消息功能块将消息主动上发给 WinCC 等 HMI 系统，适用于以下情况：

- 西门子 HMI：WinCC 或面板。
- S7 -300/400 PLC 均支持。
- 使用 ALARM_S 等块主动发送消息。
- HMI 项目必须集成在 SIMATIC Manager 中。

符号表消息适用于以下情况：

- 西门子 HMI：WinCC。
- 仅 S7 -400 PLC 支持。
- 组态符号表，无需编程。
- WinCC 项目必须集成在 SIMATIC Manager 中。

和 WinCC 的位消息相比，编号消息的优点在于：

- PLC 是基于事件主动上发消息，故总线通信负载占用少。



• 消息使用 PLC 时间戳或自定义时间戳，具有更高的时间精度。
• WinCC 更多高级的功能是基于编号消息实现的。

对于 WinCC 来说，组态符号表消息和块消息都需要满足以下条件：

• 安装 WinCC 的组件“AS－OS Engineering”。
• WinCC 项目集成到 SIMATIC Manager 中。
• 在 WinCC 项目的“计算机属性”对话框中的启动列表内激活“报警记录运行系统”。
• WinCC 版本和 STEP 7 版本必须兼容。

WinCC 项目集成到 SIMATIC Manager 中的相关概念和方法请参考本书第 3 章 3.3.5 节“WinCC 项目中直接调用 STEP 7 变量的方法”的内容。

下面分别介绍在 STEP 7 中组态符号表消息和块消息并上传到 WinCC 项目的方法。

**1. 组态符号表消息**

在 SIMATIC Manager 中，打开 PLC 程序的符号表，通过“视图”菜单选择激活“列 R，M，C，CC”，单击选中要监视的布尔型变量的 M 列的复选框，如图 5-34 所示。在符号表中用鼠标右键单击要监视的布尔型变量，在弹出的菜单中选择“特殊对象属性”→“消息”命令，打开“消息组态”对话框，如图 5-35 所示。

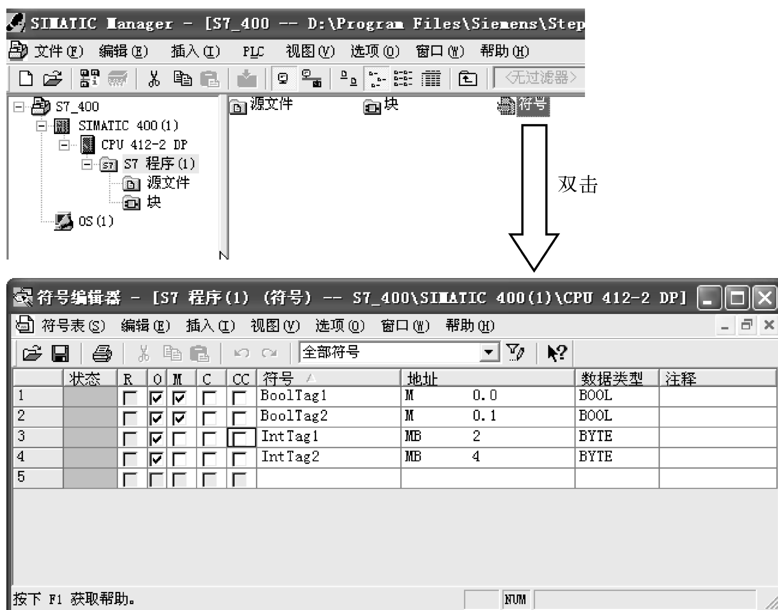


图 5-34　组态符号表消息

在“消息组态”对话框中选中“文本”选项卡，可以输入该消息相应的消息文本，选择其他的选项卡进行相关的设置后，单击对话框的“确定”按钮，保存设置并退出对话框。最后编译 OS（如图 5-34 中的 OS（1））并下载整个站到 PLC。

经过上述组态后，在 WinCC 中打开在 SIMATIC Manager 中组态的项目 OS（1），可以在 WinCC 变量管理器中找到自动生成的类型为 Raw Data 的消息帧变量（参考图 5-33），可以在 WinCC 报警记录编辑器的表格窗口找到自动生成的符号表消息，如图 5-36 所示。




图 5-35　“消息组态”对话框




图 5-36　WinCC 中生成的符号表消息

需要注意的是，只有 S7 - 400 PLC 支持符号表消息。

**2. 组态块消息**

S7 - 300/400 PLC 支持多种消息函数，表 5-13 列出了各个消息函数的区别。

**表 5-13　S7 - 300/400 PLC 支持的消息函数**

| 名称 | SFB/SFC | S7PLC | 可确认 | 通道数 | 过程值 | WinCC | OP |
|---|---|---|---|---|---|---|---|
| ALARM_SQ | SFC 17 | S7 - 300/400 | Yes | 1 | Yes | Yes | Yes |
| ALARM_S | SFC 18 | S7 - 300/400 | No | 1 | Yes | Yes | Yes |
| ALARM_SC | SFC 19 | S7 - 300/400 | - | - | - | Yes | Yes |
| ALARM_DQ | SFC 107 | 仅 S7 - 400 | Yes | 1 | Yes | Yes | Yes |
| ALARM_D | SFC 108 | 仅 S7 - 400 | No | 1 | Yes | Yes | Yes |
| NOTIFY_8P | SFB 31 | 仅 S7 - 400 | No | 8 | Yes | Yes | No |
| ALARM | SFB 33 | 仅 S7 - 400 | Yes | 1 | Yes | Yes | No |
| ALARM_8 | SFB 34 | 仅 S7 - 400 | Yes | 8 | No | Yes | No |
| ALARM_8P | SFB 35 | 仅 S7 - 400 | Yes | 8 | Yes | Yes | No |
| NOTIFY | SFB 36 | 仅 S7 - 400 | No | 1 | Yes | Yes | No |

注：某些新型的 S7 - 300 PLC 也支持 ALARM_D/ALARM_DQ，如：CPU 315 - 2 DP/PN。



下面以 SFC 107 ALARM_DQ 为例，说明块消息的组态过程：

1）在 STEP 7 中创建 FB1，在“IN”中定义 DWORD 型输入参数 EV_ID1 和 EV_ID2，并分别通过单击鼠标右键，在弹出的菜单中选择“对象属性”命令，打开“变量属性”对话框来设置两个参数的属性，如图 5-37 所示。设置参数的属性后，会在该参数上出现绿色的小旗标。




图 5-37　定义输入参数及其属性

2）在“IN”中定义 BOOL 型输入参数 Sig1 和 Sig2，在“TEMP”中定义 INT 型变量 Ret。

3）在 FB1 中调用 FC 107，其梯形图程序如图 5-38 所示。FC 107 各参数的含义如下：

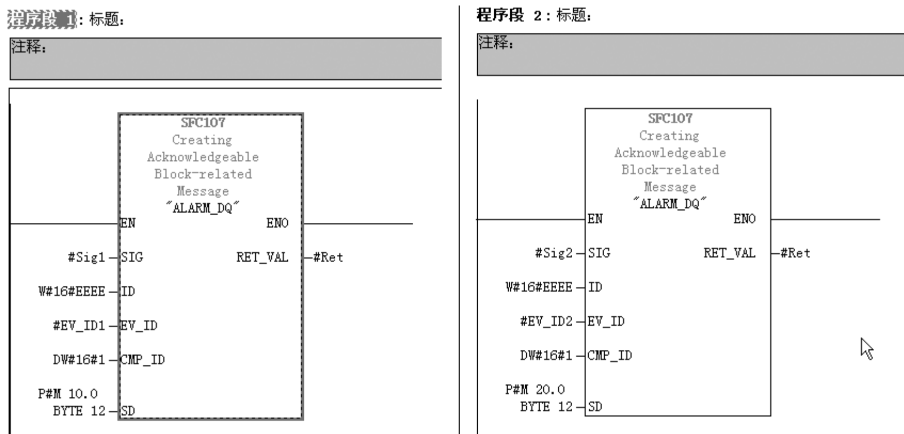



图 5-38　FB1 中调用 FC 107 的梯形图程序

- SIG 为所监视的报警变量。
- ID = W#16#EEEE。
- EV_ID 为 FB1 的输入参数 EV_ID1/EV_ID2。
- CMP_ID = DW#16#1。
- SD：为报警所携带的过程值的地址，ALARM_DQ 可携带最多 12 个字节的过程值，这些过程值可以在报警消息文本中使用。
- RET_VAL：为执行状态返回值。

4）保存并退出 FB1。打开 OB1，在 OB1 中调用 FB1。FB1 的输入参数 EV_ID1 和 EV_ID2 系统会自动分配事件编号，无需用户设置。为输入参数 Sig1 和 Sig2 指定要监视的布尔量，例如 M1.0 和 M1.1。OB1 中调用 FB1 的梯形图程序如图 5-39 所示。

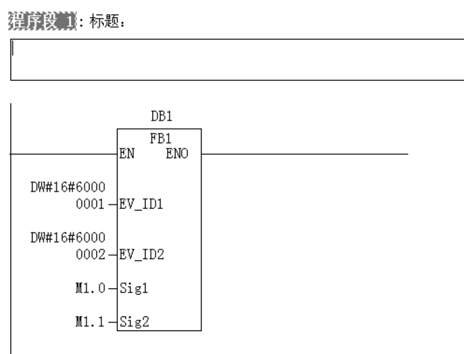



图 5-39　OB1 中调用 FB1 的梯形图程序

5）在 OB1 中选中所调用的 FB1 块，单击菜单“编辑”→“特殊对象属性”→“消息”，打开“消息组态”对话框，在对话框中组态相应的消息文本，如图 5-40 所示。消息文本中可以携带过程值，其格式为：@ PwnTpy% Width. precision Format @，具体含义见表 5-14。




图 5-40　在 OB1 中组态消息



表 5-14　消息文本携带过程值时各符号的含义

| 符　　号 | 含　　义 | 值 及 含 义 |
|---|---|---|
| @ | 起始符 | |
| Pwn | 过程值号 | 1、2、3… |
| Typ | 过程值数据类型 | b：BOOL；y：BYTE｜y；c：Chr；w：WORD；i：Int；x：DWORD；d：DINT；r：REAL |
| % width | 数据显示宽度 | |
| . precision | 数据显示精度（可选项） | |
| Format | 数据显示格式 | d：有符号十进制；u：无符号十进制；X：十六进制；f：有符号定点数；s：ANSI 字符串 |
| @ | 截止符 | |

6）保存并退出 OB1，下载整个站到 PLC。在 SIMATIC Manager 中编译 OS，结束后，打开 WinCC 报警记录编辑器，可以看到在 STEP 7 中组态的消息已经编译进入 WinCC 的消息系统，如图 5-41 所示。




图 5-41　报警记录编辑器中显示 STEP 7 组态的消息

7）在 WinCC 中新建一幅画面，在画面中添加 1 个报警控件并组态其属性。激活 WinCC，将 PLC 程序下载到 PLCSIM 并运行，设置 M1.0 和 M1.1 为 1 触发报警，如图 5-42 所示。可以在 WinCC 运行画面的消息窗口中看到相应的报警消息。




图 5-42　在 PLCSIM 设置变量值触发报警

## 5.5.4　组态在报警消息中显示系统当前登录用户的方法

消息系统可以帮助操作人员快速发现、定位并解决处理自动化系统的故障和错误，从而有效地减少停机事件和生产事故的发生。通过消息归档，消息系统还可以追溯报警和事件，进而分析事故起因、认定事故责任。在认定事故责任时，可以通过查看报警消息中显示的当前登录的用户的方法来实现。

由第4章4.6.5节“使用与用户登录相关的内部变量”可知，内部变量@ CurrentUser或@ CurrentUserName存储了WinCC项目运行时当前登录的用户的用户名信息，因此可以通过以下方法实现报警消息中显示的当前登录的用户：在组态消息系统的消息块时添加1个过程值块，过程值块的名称为“当前用户”，在组态单个消息时将该过程值块与变量@ CurrentUser建立连接，在组态WinCC报警控件时将该过程值块添加到要在消息窗口中显示的消息列中。详细组态方法和步骤如下（消息系统原有的组态数据不变）：

1）添加1个过程值块：在WinCC报警记录编辑器的浏览窗口选中“过程值块”，在数据窗口单击鼠标右键，在弹出的菜单中选择“添加/删除”命令，打开“添加过程值…”对话框。在对话框中将1个过程值块从“可用的过程值块”列表添加到“选好的过程值块”列表中。

2）修改添加的过程值块属性：在数据窗口用鼠标右键单击刚刚添加的过程值块，在弹出的菜单中选择“属性”，打开“消息块”对话框。在“名称”后的文本框中设置过程值块的名称为“当前用户”。“长度”设置为10字符。

3）组态单个消息：在报警记录编辑器的表格窗口用鼠标右键单击需要显示当前登录用户信息的消息行，在弹出的菜单中选择“属性”命令，打开“单个消息”对话框。在对话框中选择“变量/动作”选项卡，单击“过程值”框内“当前用户”后的按钮[...]，打开“变量-项目”对话框，选择变量@ CurrentUser，如图5-43所示。单击“确定”按钮退出对话框。



图5-43 建立过程值块与变量@ CurrentUser的连接

4）组态WinCC报警控件：在WinCC图形编辑器中打开WinCC报警控件所在画面，双击WinCC报警控件，打开“WinCC报警控件属性”对话框。在对话框中选择“消息列”选项卡，将“当前用户”过程值块从“现有消息块”列表添加到“选定消息块”列表中。单击“确定”按钮退出对话框。

5）在图形编辑器中保存所作的组态，运行WinCC项目，可以在消息窗口中新出现的消息看到当前登录用户的信息（必须组态了用户管理且在WinCC项目运行后有用户登录），如图5-44所示。



面中显示用户信息

用户管理的组态以及用户登录 WinCC 运行系统的相关概念请参考本书第 4 章 4.6 节的相关内容。

## 5.5.5 通过 C 脚本确认报警消息的组态方法

如果在 WinCC 运行系统中不希望通过消息窗口的工具栏，而是通过其他方式例如单击画面中的按钮来操作 WinCC 报警控件，可以通过编程调用 C 脚本编辑器的“报警”组提供的多个可用的标准函数来实现，这些标准函数的功能描述见表 5-15。

表 5-15 报警相关的标准函数

| 函数 | 描述 |
|---|---|
| AXC_OnBtnAlarmHidingList | 使用该函数切换消息窗口以显示已隐藏消息列表 |
| AXC_OnBtnArcLong | 使用该函数切换消息窗口以显示长期归档列表中的消息 |
| AXC_OnBtnArcShort | 使用该函数切换消息窗口以显示短期归档列表中的消息 |
| AXC_OnBtnComment | 此函数打开对话框，以便输入消息注释 |
| AXC_OnBtnEmergAckn | 此函数用于紧急确认消息 |
| AXC_OnBtnHideUnhideMsg | 该函数用于隐藏所选的消息或再次显示隐藏的消息 |
| AXC_OnBtnHit | 使用该函数切换消息窗口以显示统计列表中的消息 |
| AXC_OnBtnHornAckn | 此函数可用于确认分配给所选消息的中央信令设备 |
| AXC_OnBtnInfo | 该函数打开一个信息窗口，显示所存储的信息文本 |
| AXC_OnBtnLoop | 使用此函数将所显示的画面变为所选消息的已组态报警回路画面 |
| AXC_OnBtnMsgFirst | 此函数选择第一条消息并在必要时改变消息窗口的可见范围 |
| AXC_OnBtnMsgLast | 此函数选择最后一条消息并在必要时改变消息窗口的可见范围 |
| AXC_OnBtnMsgNext | 此函数选择下一条消息并在必要时改变消息窗口的可见范围 |
| AXC_OnBtnMsgPrev | 此函数选择前一条消息并在必要时改变消息窗口的可见范围 |
| AXC_OnBtnMsgWin | 使用该函数切换消息窗口以显示消息列表中的消息 |
| AXC_OnBtnPrint | 根据当前的显示类型（消息列表、短期归档列表和长期归档列表），使用该函数生成满足选择标准的当前未决消息或已归档消息的打印输出。打印布局取决于消息窗口的类型 |
| AXC_OnBtnProtocol | 此函数启动控件的当前视图的打印输出。打印所有满足选择标准的消息 |
| AXC_OnBtnScroll | 此函数用于改变消息窗口的自动滚动行为 |
| AXC_OnBtnSelect | 此函数打开选择对话框 |
| AXC_OnBtnSinglAckn | 此函数用于确认所选的单个消息 |
| AXC_OnBtnSortDlg | 该函数打开对话框，以便创建所显示消息的自定义排序标准 |
| AXC_OnBtnTimeBase | 该函数打开对话框，以便为消息中显示的时间数据设置时间基准 |
| AXC_OnBtnVisibleAckn | 此函数确认消息窗口中所有可见的消息 |
| AXC_OnBtnLock | 此函数打开对话框，以便锁定消息 |
| AXC_OnBtnLockWin | 此函数激活锁定列表窗口 |

（续）

| | |
|---|---|
| AXC_OnBtnLockUnlock | 在消息列表视图、短期归档列表视图和长期归档列表视图中，使用该函数锁定当前所选择的报警控件消息。如果选择锁定列表视图，使用此函数解锁当前所选择的消息 |
| AXC_SetFilter | 此函数用于定义过滤器，用来选择要在消息窗口中显示的消息。必须在动作脚本中定义过滤标准 |

可以在 WinCC 过程画面中直接组态按钮的 C 动作就可以实现“确认报警信息”、“显示过滤”、“紧急确认” 等操作。以确认报警窗口中所选的单个消息为例，组态方法和步骤如下：

1）打开 WinCC 报警控件所在的画面，从对象选项板的“标准”→“窗口对象”，列表中选择一个“按钮” 添加到画面的合适位置，并在自动弹出的“按钮组态” 对话框中设置该按钮的“文本” 属性为“确认单个消息”。

2）打开按钮的“对象属性” 对话框，选择“事件”→“鼠标”→“按左键”，为“按左键” 的动作组态 C 动作，动作的 C 代码程序如下：

```
void OnLButtonDown(char * lpszPictureName, char * lpszObjectName, char * lpszPropertyName,
                    UINT nFlags, int x, int y)
{
AXC_OnBtnSinglAckn (lpszPictureName," 控件 1");
}
```

程序中“控件 1” 即 WinCC 报警控件的对象名。如果组态确认消息窗口中所有可见的报警的按钮，只需要将函数 AXC_OnBtnSinglAckn 变为 AXC_OnBtnVisibleAckn 即可。

# 第6章

# 变量记录系统

变量记录亦称为变量归档或过程值归档，主要目的是采集、处理和归档工业现场的过程数据。这些数据有助于提前检测危险和故障条件，从而提高生产力和产品质量、优化维护周期等。通过过程值的归档，可以对工业现场的历史生产数据做一个完备的记录，从而对当前和今后的工作起到很强的借鉴作用。

本章将详细介绍 WinCC 变量记录系统对变量进行归档的方法，以及在运行系统中以趋势曲线和表格的方式显示归档的历史数据的方法。

## 6.1 WinCC 变量记录系统的相关概念

本节将介绍 WinCC 变量记录系统的一些相关概念，为熟悉和掌握 WinCC 变量记录系统的组态打下基础。

### 6.1.1 过程值归档的原理

需要归档的过程值在归档系统中进行编译、处理，并在运行系统数据库中暂存，然后写到归档数据库中，整个工作流程如图 6-1 所示。

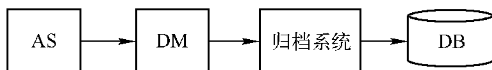



图 6-1 过程值归档的流程

过程值归档涉及下列 WinCC 子系统：

- 自动化系统（AS）：存储通过通信驱动程序传送到 WinCC 的过程值。
- 数据管理器（DM）：处理过程值，然后通过过程变量将其返回到归档系统。
- 归档系统：处理采集到的过程值（例如，产生平均值）。处理方法取决于组态归档的方式。
- 运行系统数据库（DB）：保存要归档的过程值。

**1. 归档变量的类型**

过程值存储在归档的归档变量中，WinCC 变量记录系统区分以下类型的归档变量：

- 模拟量归档变量存储数字过程值，例如罐的填充量。
- 二进制归档变量存储二进制过程值，例如电机的打开或关闭。

- 过程控制变量存储作为报文发送到归档系统的过程值，例如在一系列测量中的过程值。这种变量在 WinCC 中定义为“原始数据类型”，因此也可称为“原始数据变量”。

如果希望对采集的原始数据变量的过程值进行归档，则需要在过程值归档中组态一个过程控制变量。为使归档系统能够处理过程控制变量中的报文，选择格式 DLL。这样，将为用户正在使用的自动化系统提供格式 DLL，并由后者拆分报文（例如以确定过程值）。然后将过程值写入归档数据库。

WinCC 变量记录系统还可压缩已归档的过程值，该压缩通过应用数学函数（例如求平均值）来实现，该类型的压缩过程值存储在压缩归档的压缩变量中。

WinCC 基本版包含 512 可使用的归档变量。更多的归档组态需要升级扩展包。

**2. 周期和事件**

过程值归档用周期和事件进行控制。采集和归档周期可确保过程值的连续采集和归档，也可由事件和动作启动或结束过程值归档。此外，还可以组合使用周期和事件，例如，可定期采集一个过程值，却由二进制事件启动归档。

（1）采集周期

采集周期确定读取过程变量过程值的时间间隔，最短可以是 500 ms，所有其他值是采集周期的整数倍。由 WinCC 运行系统的启动时间确定采集周期的起始点。

需要注意的是，较短的采集周期可能导致较高的系统负载，在采集频繁或快速改变的过程值时，建议使用原始数据变量。

（2）归档周期

归档周期确定何时将过程值保存到归档数据库中，归档周期总是采集周期的整数倍。归档周期的起始点取决于 WinCC 运行系统的启动时间或所使用时间的起始点。

需要说明的是，如果采集和归档使用相同的周期，那么这并不一定表示同时启动采集和归档。在采集和归档之间可能相差高达一个采集周期。

（3）归档函数

在采集和归档周期之间从过程变量中读取的所有过程值都将由归档函数进行处理。在过程值归档中，可以使用下列归档函数中的一种：

- 当前值：保存所采集的最后一个过程值。
- 总值：保存所有采集到的过程值的总和。
- 最大值：保存所有采集的过程值的最大值。
- 最小值：保存所有采集的过程值的最小值。
- 平均值：保存所有采集到的过程值的平均值。
- 动作：最近采集的过程值由全局脚本中创建的函数进行计算。

（4）启动/停止事件和动作

可用事件启动和停止过程值归档，触发事件的条件可以链接到变量或脚本（C 或 VBS）。在 WinCC 中，分为以下类型的事件或动作：

- 二进制动作：对布尔类型的过程变量的改变作出响应。例如，接通电机可以启动过程值归档。
- 限制值事件：对超过或达到一个限制值作出响应，可以用绝对值或比例值来表示限



制值事件。例如，温度波动超过 2% 触发归档。

- 时间控制事件：在固定时间点或启动过程值归档后经过一定的时间作出响应。例如，每次换档改变后，做一个记录。

## 6.1.2 归档方法

WinCC 变量记录系统中，是否以及何时采集和归档过程值取决于各种参数，组态哪些参数取决于所使用的归档方法，WinCC 可以使用不同的归档方法来归档过程值。例如，用户可以在任意时间监控单个过程值并使该监控依赖于某些事件。可以快速归档变化的过程值，而不会导致系统负载的增加。用户还可以压缩已归档的过程值来减少数据量。

在 WinCC 运行系统中可用下列归档方法：

- 过程值的连续周期性归档：连续的过程值归档。例如监控一个过程值。
- 周期的选择性过程值归档：受事件控制的连续的过程值归档。例如，用于在特定时段内对某过程值进行监视。
- 非周期性的过程值归档：事件控制的过程值归档。例如，当超出临界限制值时，对当前过程值进行归档。
- 在改变期间将过程值归档：过程值仅在被改变后才可进行非周期性归档。
- 过程控制的过程值归档：对多个过程变量或快速变化的过程值进行归档。
- 压缩归档：压缩单个归档变量或整个过程值归档。例如，对每分钟归档一次的过程值求每小时的平均数。

**1. 过程值的连续周期性归档**

运行系统启动时，过程值的连续周期性归档也随之开始。过程值以恒定的时间周期采集并存储在归档数据库中。运行系统终止时，过程值的连续周期性归档也随之结束。

**2. 周期的选择性过程值归档**

发生启动事件时，在运行系统中开始周期的选择性过程值归档。过程值以恒定的时间周期采集并存储在归档数据库中。

周期性过程值归档会在发生以下情况时结束：

- 发生停止事件时。
- 终止运行系统时。
- 启动事件不再存在时。

**3. 非周期性的过程值归档**

在系统运行期间，非周期性过程值归档仅将当前过程值保存在归档数据库中。在以下情况下才进行归档：

- 在每次更改过程值时。
- 在每次启动和结束启动事件时（如果组态了启动事件）。

终止运行系统时，非周期性的过程值归档结束。

**4. 过程控制的过程值归档**

在过程控制的过程值归档中，将要归档的过程值在 PLC 块中编块，通过 WinCC 项目

管理器将其作为原始数据变量发到变量记录中。使用转换程序、格式化 DLL 在变量记录中准备数据并且存储在归档中。格式化 DLL 时由通道决定的，因此必须遵守通道和 PLC 生产商的技术规范。变量记录系统提供用于该目的接口的定义。

5. 压缩归档

为了减少归档数据库中的数据量，可以对指定时期内的归档变量进行压缩归档。为此，需要创建一个压缩归档，将每个归档变量存储在压缩变量中。归档变量将保留，但是也可以对其进行复制、移动或删除等操作。压缩归档以与过程值归档相同的方式存储在归档数据库中。压缩通过数学函数实现，为此，在指定时间段内，下列函数之一被应用于归档过程值：

- 最大值：将最大过程值存储在压缩变量中。
- 最小值：将最小过程值存储在压缩变量中。
- 平均值：将过程值的平均值存储在压缩变量中。
- 总和：将过程值的总和存储在压缩变量中。

原来的归档过程值在压缩后如何处理取决于所使用的压缩方式：

- 计算：指定时期内的归档变量过程值被读出并压缩。归档变量的过程值被保留。
- 计算并复制：指定时期内的归档变量的过程值被读出、压缩并复制到压缩归档。
- 计算并删除：指定时期内的归档变量过程值被读出、压缩然后删除。
- 计算、复制并删除：指定时期内的归档变量的过程值被读出、压缩并移动到压缩归档。

### 6.1.3　过程值的存储

1. 存储方式

过程值可存储在归档数据库的硬盘上，或存储在变量记录运行系统的主存储器中。

（1）存储在归档数据库中

要归档的过程值存储在归档数据库的两个独立的循环归档（A、B）中，如图 6-2 所示。循环归档由数目可组态的数据缓冲区组成。数据缓冲区根据大小（以 MB 计）和时间周期（例如一天）定义。

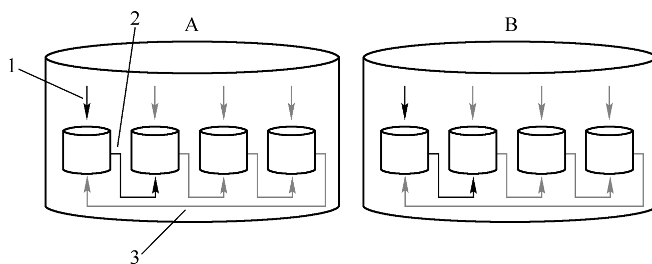



图 6-2　过程值在归档数据库中的存储

过程值被连续写入数据缓冲区中，如图 6-2 中箭头 1 所示。如果达到数据缓冲区所组态的大小或超出时间段，系统切换到下一个数据缓冲区，如图 6-2 中箭头 2 所示。当所有数据缓冲区满时，第一个数据缓冲区中的过程数据会被覆盖，如图 6-2 中的箭头 3



所示。为了使过程数据不被覆盖过程破坏，可以将其先导出。

循环归档 A 存储采集周期短于或等于一分钟的过程值（快速归档）。这些过程值最初保存并压缩在一个二进制文件中。当二进制文件达到指定大小时，将其存储到循环归档中。

循环归档 B 存储采集周期大于一分钟的过程值和压缩归档（慢速归档）。数据被立即写入循环归档中并且不被压缩。

（2）保存在主存储器中

与存储在归档数据库中不同，在主内存中归档的过程值只在系统激活时有效。然而，存储在主内存中的优点是可以非常快速地写入和读出数据。存储在主内存中的过程值无法换出。

压缩归档无法存储在主内存中。

**2. 运行数据的存储文件**

运行数据分别存储于数据库母体和几个运行数据库中。数据库母体管理运行数据库，数据库母体创建在项目的文件夹下，名字由项目名称和“RT”尾标构成。每个运行数据库包含特定时间段的归档数据，存储于项目目录下的子目录“ArchiveManager/ TagLoggingFast or TagLoggingSlow ”。

运行数据库的名称构成：计算机名 + 项目名 + 数据类型的 ID（ALG = Alarm Logging,, TLG_F = Tag Logging Fast,, TLG_S = Tag Logging Slow） + 日期/时间（年，月，日，时，分）。

（1）运行数据的快速变量记录文件

- 计算机名_项目名_TLG_Fast_ <起始时间>_ <结束时间>. LDF。
- 计算机名_项目名_TLG_Fast_ <起始时间>_ <结束时间>. MDF。
- 计算机名_项目名_TLG_Fast_YYYYMMDDHHMM. MDF（不完整记录）。

例如：SATCB_WinCCPjtv60_TLG_F_200306020629. MDF。

SATCB_WinCCPjtv60 _TLG_F_200306020901_200306021000. MDF。

（2）运行数据的慢速变量记录文件

- 计算机名_项目名_TLG_Slow_ <起始时间>_ <结束时间>. LDF。
- 计算机名_项目名_TLG_Slow_ <起始时间>_ <结束时间>. MDF。
- 计算机名_项目名_TLG_Slow_YYYYMMDDHHMM. MDF（不完整记录）。

例如：SATCB_WinCCPjtv60 _TLG_S_200306020629. MDF。

SATCB_WinCCPjtv60 _TLG_S_200306020901_200306021000. MDF。

## 6.2 组态过程值归档

变量记录系统分为组态系统和运行系统两个部分。组态系统的任务是在 WinCC 变量记录中定义归档、归档变量和归档周期等，可以使用归档向导简化变量归档的组态。运行系统的任务是归档过程变量，并借助在图形编辑器中组态的 WinCC Online Table Control 或 WinCC Online Trend Control 这两个控件，以不同的形式显示归档的过程变量。

### 6.2.1　组态变量记录系统的步骤

WinCC 项目中组态变量记录系统包括如下的步骤：

1）在变量记录中创建和配置用于变量归档的定时器，可以自定义定时器，也可以直接使用默认定时器。

2）在变量记录中使用归档向导创建和配置一个过程值归档，用于存储过程数据。

3）如果有必要，在所创建的归档中对每个归档变量进行属性配置。

4）在图形编辑器中创建和配置 WinCC Online Table Control 或 WinCC Online Trend Control，以便在系统运行时观察归档数据。

### 6.2.2　WinCC 变量记录

可在变量记录中对归档、要归档的过程值以及采集时间和归档周期等进行组态。此外，还可以在变量记录中定义硬盘上的数据缓冲区以及如何导出数据。

在 WinCC 项目管理器的浏览窗口双击“变量记录”，或用鼠标右键单击“变量记录”，在弹出的菜单中选择“打开”命令，即可打开 WinCC 变量记录，如图 6-3 所示。



图 6-3　WinCC 变量记录

WinCC 变量记录分为浏览窗口、数据窗口和表格窗口：

- 浏览窗口：此处选择是否想要编辑定时器或归档。
- 数据窗口：根据在浏览窗口中所作的选择，可在此处编辑已存在的归档或定时器，或者创建新的归档或定时器。
- 表格窗口：此处是显示归档变量或压缩变量的地方，这些变量存储于在数据窗口中所选的归档中。可以在此改变显示的变量的属性或添加一个新的归档变量或压缩变量。

### 6.2.3　组态定时器

如果要用于采集和归档周期的定时器偏离了标准定时器，可组态新的定时器。新的



定时器以基数（例如，1 秒）的整数倍计算。组态新的定时器的步骤如下：

1）在变量记录的浏览窗口中，用鼠标右键单击“定时器”，在弹出的菜单选择“新建...”命令，打开“定时器属性”对话框，如图 6-4 所示。




图 6-4　“定时器属性”对话框

2）定义新的定时器：输入名称，例如“6 秒”。从“基准”列表中选择期望的基数，例如“1 秒”；然后在“系数”框中输入整数因子，例如“6”。

3）可在“循环起始点”中为新的定时器指定更多的属性。通过输入新的起始点，可指定何时由周期执行归档（例如在每 6 秒的第 0 秒执行）。起始点规范可使归档负载均匀分布。

4）单击“确定”按钮应用新的定时器。它将显示在数据窗口中，并可选择作为采集或归档周期。

### 6.2.4　在变量记录中创建归档

在创建归档时，以下归档类型之间具有区别：

- 过程值归档存储归档变量中的过程值。在组态过程值归档时，选择要归档的过程变量和存储位置。
- 压缩归档压缩来自过程值归档的归档变量。在组态压缩归档时，选择计算方法和压缩时间段。

**1. 通过归档向导创建过程值归档**

在 WinCC 变量记录中通过归档向导创建过程值归档的步骤如下：

1）在变量记录的浏览窗口中，用鼠标右键单击“归档”，在弹出的菜单中选择“归档向导...”命令，打开“创建归档”对话框。

2）在“归档名称”框中输入合适的归档名称，选择归档类型为“过程值归档”。单击“下一步”按钮。

3）单击“选择”按钮，打开“变量 - 项目对话框”，以选择想要进行归档的变量。

也可在创建过程值归档之后添加其他变量。

4）单击“完成”按钮，新的过程值归档包含在变量记录的数据窗口内，所选择的归档变量显示在表格窗口中。

**2. 通过归档向导创建压缩归档**

在 WinCC 变量记录中通过归档向导创建压缩归档的步骤如下：

1）在变量记录的浏览窗口中，用鼠标右键单击“归档”，在弹出的菜单中选择“归档向导...”命令，打开“创建归档”对话框。

2）在“归档名称”框中输入适当的归档名称，选择归档类型为“压缩归档”，单击“完成”按钮，新的压缩归档包含在变量记录的数据窗口内。

### 6.2.5　创建并组态归档变量

前文已述及，归档变量用来存储要归档的过程值，下列归档变量可在过程值归档中使用：

- 二进制归档变量用于存储二进制过程值。
- 模拟量归档变量用于存储数字过程值。
- 过程控制变量用于存储作为报文发送到归档系统的过程值。

在压缩归档中，每个压缩的过程值都存储在一个单独的压缩变量中。

**1. 创建归档变量的基本步骤**

创建归档变量的基本步骤如下：

1）在变量记录的数据窗口中，选择想要在其中创建新归档变量的归档，单击鼠标右键，在弹出的菜单中选择“新建变量”命令，打开“变量 - 项目”对话框。

2）在“变量 - 项目”对话框中，选择想要对其值进行归档的变量（二进制变量或模拟量变量）。

3）单击“确定”按钮接受所选的变量。

创建的归档变量将在表格窗口中显示。

**2. 组态二进制归档变量**

在按照创建归档变量的基本步骤创建二进制归档变量后，可以对其进行归档组态。

（1）组态二进制归档变量的属性

1）在变量记录的数据窗口中选择过程值归档，在表格窗口中，用鼠标右键单击二进制归档变量，在弹出的菜单中选择“属性”命令，打开“过程变量属性”对话框，如图 6-5 所示。

2）在“归档变量”选项卡中，根据需要修改归档变量名称，也可与过程变量名称相同。

3）定义归档变量是手动输入还是由系统提供。

4）在“归档”组中，定义是否在系统启动时激活归档。如果中央归档服务器（CAS）将归档变量视为具有长期关联性的变量，则选中“相关长期”的复选框。

5）还可将归档变量值写入到内部变量中，以便将归档变量用于其他用途。更新过程取决于归档变量的周期设置。单击“同时在变量中输入归档值”组内的“选择”按钮，打开“变量 - 项目”对话框来选择内部变量。






图 6-5　“过程变量属性”对话框

（2）组态归档方法和采集类型

1）在“过程变量属性”对话框的“归档”选项卡中选择归档方法和采集类型，可组态过程值的连续周期性归档、有选择的周期性归档、事件控制的非周期性归档和过程值修改时的非周期性归档四种归档方法。

2）对于连续周期性归档，如图 6-6 所示，在“归档类型”下拉列表框中选择“周期”归档类型，并在“归档”组内设置归档周期。




图 6-6　选择“周期”归档方法

3）对于有选择的周期性归档，还可在“动作”组中组态开始和停止事件，单击...按钮，在变量管理器中选择一个变量或在函数浏览器中选择一个 C 脚本。该变量或函数必须返回“TRUE”或“FALSE”值。如果出现 TRUE 值，则归档将随可能的情况启动或停止。

4）对于事件控制的非周期性归档，如图 6-7 所示，在“归档类型”下拉列表框中选择“非周期”归档类型，在“采集”下拉列表中选择“事件控制”。需要在“事件”组中定义基本事件。单击...按钮，在变量管理器中选择一个变量或在函数浏览器中选择一个 C 脚本。该变量或函数必须返回“TRUE”或“FALSE”值，如果出现“TRUE”值，则归档启动。

5）若要在值改变时将过程变量归档，如图 6-8 所示，请选择归档方法为“非周期”和采集类型为“改变时”。在 WinCC 控件中，定义一个显示周期以显示长时间未修改的值。否则，该值最后将从显示中消失。在区段更改期间，如果在变量的过程值未更改的情况下还要归档该值，需要选中“区段变化后归档”的复选框。




图 6-7　选择“非周期”归档方法“事件控制”采集类型

（3）归档参数

1）在“过程变量属性”对话框中选择“参数”选项卡，在“归档”组内指定何时归档过程值。如果当前过程值始终要在趋势图中显示，则选择“一直”。

2）在“出错误时保存”组中，定义可接受的错误值。

3）在为有选择的周期性归档组态了启动和停止动作后，可在“数值数量”组中输入头值和尾值。这些动作在采集周期中调用和编辑，具体取决于应用程序的头部或尾部缓冲区。

4）通过单击“确定”按钮结束二进制归档变量的组态。






图 6-8　选择“非周期”归档方法“变化时”采集类型

### 3. 组态模拟量归档变量

在按照创建归档变量的基本步骤创建模拟量归档变量后，可以对其进行归档组态。

（1）组态模拟量归档变量的属性

在变量记录数据窗口中选择过程值归档，在表格窗口中，用鼠标右键单击模拟量归档变量，然后在弹出的菜单中选择“属性”菜单项，打开“过程变量属性”对话框，组态模拟量归档变量的属性的操作与组态二进制归档变量的属性的操作类似，在此不再赘述。

（2）归档方法和采集类型

模拟量归档变量的归档方法和采集类型的组态与二进制归档变量的归档方法和采集类型的组态类似，不再赘述。

（3）归档参数

1）如图 6-9 所示，对于周期性归档，在“参数”选项卡的“处理”组中选择一个归档函数。如果选择的是“动作”选项，则通过“全局脚本”创建的函数计算最近记录的过程值，此时通过 选择... 按钮调用函数浏览器。

2）在组态了启动和停止动作后，可在“数值数量”组中输入头值和尾值。这些动作在采集周期中调用和编辑，具体取决于应用程序的头部或尾部缓冲区。

3）在“出错时保存”组中，定义可接受的错误值。

（4）模拟量归档变量的显示区域或添加区域

如图 6-10 所示，要记录仅归档或显示介于下限和上限之间的过程值，在“过程变量属性”的“显示”选项卡中选择“直接组态”，并在“下限”和“上限”框中分别输入下限和上限值。




图 6-9　组态模拟量归档变量的归档参数




图 6-10　组态模拟量归档变量的显示区域或添加区域

**4. 创建过程控制变量**

前文已述及，为采集和归档快速改变的过程值或来自多个测量点的过程值，使用原始数据变量将数值从自动化系统发送到 WinCC。为了对归档变量中的数据进行归档，将原始数据变量分配到归档变量的过程需要一个过程控制变量进行定义。只有为归档变量提供数据的原始数据变量可在过程控制变量中分配。

创建过程控制变量的步骤如下：



1）在变量记录的数据窗口中，选择想要在其中创建新的过程控制变量的过程值归档。然后在表格窗口中单击鼠标右键，并在弹出的菜单中选择“新建过程控制变量”菜单项，打开“过程控制变量属性”对话框，如图 6-11 所示。

2）从“转化 DLL”组内的下拉列表中，选择与正在使用的控制器匹配的“格式 DLL”。

3）单击 选择... 按钮，在变量选择对话框中选择原始数据变量。当选择了原始数据变量时，会提示用户输入一个或多个 ID 号。ID 号的数量取决于所选择的格式 DLL，ID 号的数值用于形成内部归档变量名。单击 选项... 按钮可以编辑 ID 号。

4）如果需要，在“归档变量名”下输入过程控制变量的别名。如果没有在该域内输入名称，则使用 WinCC 中的内部归档变量名。

5）如果中央归档服务器（CAS）将归档变量视为具有长期关联性的变量，则激活控件框“相关长期”。

6）单击“确定”按钮，接受新的过程控制变量。




图 6-11　“过程控制变量属性”对话框

### 5. 创建压缩变量

创建压缩变量的步骤如下：

1）在变量记录的数据窗口中选择压缩归档。然后在表格窗口中单击鼠标右键，并在弹出的菜单中选择“选择变量”菜单项，打开“选择压缩变量”对话框，如图 6-12 所示。

2）所有可用的归档显示在窗口的左半部分。

3）要将归档变量包含在压缩归档中，选择归档变量，然后单击“ > ”按钮。如果想要包括所有显示的归档变量，单击“ >> ”按钮。

4）单击“确定”按钮以传送压缩变量。每一个所选归档变量的压缩变量在表格窗口中显示。

创建压缩变量后，可以对压缩变量进行组态，步骤如下：

1）在变量记录的数据窗口中选择压缩归档，在表格窗口中单击鼠标右键，然后从弹出的菜单中选择“属性”菜单项，打开“压缩变量属性”对话框，如图 6-13 所示。

2）如果需要，在“归档变量”选项卡的“归档变量名称”框中改变压缩变量的名称，不改变则与过程变量名称相同。

3）在“归档变量”选项卡的“归档”组中，定义是否在系统启动时激活归档。如果中央归档服务器（CAS）将归档变量视为具有长期关联性的变量，则选中“相关长期”的复选框。

4）在“参数”选项卡的“处理”组中，选择压缩过程值的数学函数。

5）单击“确定”按钮结束压缩变量的组态。




图 6-12　“选择压缩变量”对话框




图 6-13　“压缩变量属性”对话框

## 6.2.6　组态归档

在 WinCC 变量记录中组态归档，归档周期小于 1 分钟的归档的数据缓冲区在“高速变量记录”（TagLogging Fast）中组态。归档周期大于 1 分钟的归档的数据缓冲区在“低速变量记录”（）中组态。组态步骤如下：

1）在 WinCC 变量记录的浏览窗口中选择“归档组态”，在数据窗口中，双击“TagLogging Fast”或“TagLogging Slow”，打开“TagLogging Fast”或“TagLogging Slow”对话框，如图 6-14 所示。

2）在对话框的“归档组态”选项卡中为归档作下列设置：

- 所有分段的时间段以及最大尺寸。这个规范定义了归档数据库的大小，如果超出其中一个标准，则启动新的分段并删除最旧的分段。
- 单个分段所包含的时间段以及最大尺寸。如果违反了其中任意一个限制，则将启动一个新的单个分段。如果超出“所有分段的时间段”标准的限制，则最早的单个分段将被删除。



3）在“更改分段的时间”中输入下列信息：首次改变段的开始日期和开始时间。

4）单击“确定”按钮以确认输入。

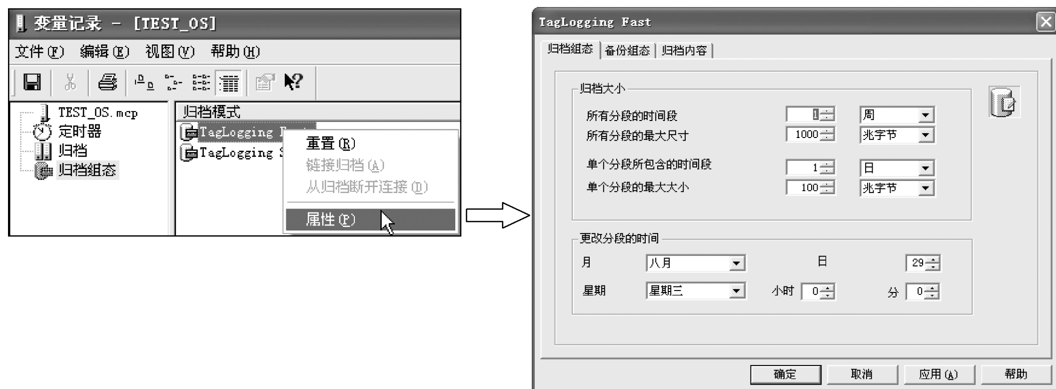



图 6-14　组态归档

如果用户不改变各个设置，那么在“高速变量记录”过程值归档中将以不大于 1 分钟的周期时间归档归档变量。周期时间大于 1 分钟的归档变量将在“低速变量记录”过程值归档中进行归档。在运行系统中，禁止将归档变量的周期改变为上述限制之外的值。不能在运行系统中将归档变量的归档类型从“快速”改变为“低速”，反之亦然。

周期改变或重新组态后，如果要以不同的归档类型存储变量时，则将从当前有效的归档中读取所述变量。不能在运行系统中访问该变量的以前归档值。改变后，当这些变量不再在“高速变量记录”中而在“低速变量记录”中归档变量时，这些变量所需数据库的大小可能会增大好几倍。

## 6.2.7　组态归档备份

创建过程值归档数据的定期备份，可以确保过程值的文档没有间隔。

通常在与时间相关的分段首次改变 15 分钟后开始备份，如果备份启动和分段启动应和运行系统同步启动，在系统运行前定义分段改变的启动时间。

可通过“在线表格控件”更改运行系统中所显示的过程值。如果在其中存储过程值的归档段的位置已发生变化，则在改变后的归档中将不接受已更改值，更改仅在本地归档段中完成。如果归档段尚未移出，则已更改值将被永久接受。

### 1. 归档备份的组态步骤

组态归档备份的步骤如下：

1）在 WinCC 变量记录的浏览窗口中，选择“归档组态”，在数据窗口中，双击“TagLogging Fast”或“TagLogging Slow”，打开“TagLogging Fast”或“TagLogging Slow”对话框。

2）选择“备份组态”选项卡。

3）如果归档备份文件将包含签名，那么选中“激活签名”的复选框。与 WinCC 重新连接时，通过签名可使得系统能确定在交换后是否修改了归档备份文件。

4）如果要备份归档数据，则选中“激活备份”的复选框。如果要在“目标路径”和“备选目标路径”两个目录下均保存已归档的数据，那么选中“备份到两个路径”的复选框。

5）在“目标路径”和“备选目标路径”的文本框中输入存储备份文件的目标路径，网络路径也可作为目标路径。在下列情况下使用“备选目标路径”：

- 在备份介质上的存储空间已满。
- 原始目标路径不可用，例如，由于网络出现故障。

6）单击“确定”按钮确认输入。

**2. 归档备份的链接**

为了可以访问运行系统归档备份中的文件，将相关的数据库文件重新连接到项目，可使用 WinCC 变量记录手动链接到归档，或自动建立链接。链接归档备份时，归档备份的相应的 LDF 文件和 MDF 文件位于组态计算机的本地目录下，例如硬盘、磁盘（MOD）或光盘（CD），对应的 WinCC 项目装载到组态计算机上，并处于运行状态。

（1）使用变量记录编辑器链接到归档

1）在 WinCC 项目管理器中打开变量记录。

2）在变量记录的浏览窗口中选择“归档组态”，在数据窗口中，用鼠标右键单击“TagLogging Fast”或“TagLogging Slow”，然后从弹出的菜单中选择“链接归档”。

3）选择所需的数据库文件并单击“确定”按钮。

数据库文件与项目链接，过程值可以直接在运行系统中显示。如果激活签名后将已修改或未签名的归档备份文件链接到项目，则必须确认要链接到这些文件，否则不会链接到归档备份文件，同时生成一条 WinCC 系统消息，并在 Windows 事件日志的“应用程序”部分中添加一个条目。

连接“低速变量记录”归档将导致验证时间较长，“链接归档”菜单条目在连接期间显示为灰色阴影。

（2）自动链接到归档

1）将归档备份文件添加到“ProjectName \ CommonArchiving”目录下。

2）在运行系统时，过程值归档自动链接到项目。

如果已激活签名，那么已修改签名的归档备份文件将不会自动链接，将生成一条 WinCC 系统消息，并在 Windows 事件日志的“应用程序”部分中添加一个条目。

**3. 断开归档备份**

如果在运行系统时不再需要访问归档备份中的数据，那么可从项目断开相关的数据库文件，步骤如下：

1）在 WinCC 项目管理器中打开变量记录。

2）在变量记录的浏览窗口中选择“归档组态”条目，在数据窗口中，用鼠标右键单击“TagLogging Fast”或“TagLogging Slow”，然后从弹出的菜单中选择“断开归档”。

3）选择所需的归档文件并单击“确定”按钮。

断开归档文件的链接后，在运行系统不能再访问其归档过程值。

## 6.3　过程值归档的输出

WinCC 图形编辑器提供两个 ActiveX 控件用于显示过程值归档，WinCC Online Trend Control 以趋势的形式显示已归档的过程变量的历史值和当前值，则 WinCC Online Table



Control 以表格的形式显示。

## 6.3.1 以趋势的形式输出过程值

### 1. 趋势的显示类型

WinCC 有 3 种基本趋势类型用于图形显示过程变量的值，其示意图如图 6-15 所示。

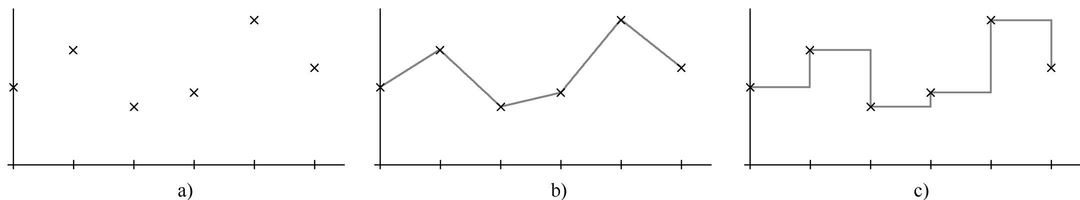



图 6-15 趋势显示的三种类型

a）单点数值 b）线性插补 c）步进趋势

- 单点数值：测量点的值显示为点。
- 线性插补：将趋势线从测量点的值进行线性插补，趋势将显示为一条实线或虚线，曲线下面的区域也可使用单色显示。
- 步进趋势：将趋势线作为步进曲线从测量点的值进行插补，趋势将显示为一条实线或虚线，曲线下面的区域也可使用单色显示。

### 2. 在趋势窗口中显示多个趋势

如果要在趋势窗口中显示多个趋势，可以选择每个趋势使用单独的坐标轴，也可以选择所有趋势使用公共的 X/Y 轴。

（1）使用不同坐标轴显示多个趋势

如果要在趋势窗口中显示的变量值差别很大，建议不要使用公共坐标轴显示趋势。使用不同的坐标刻度，读取变量值会更清晰、更容易一些，如图 6-16 所示。

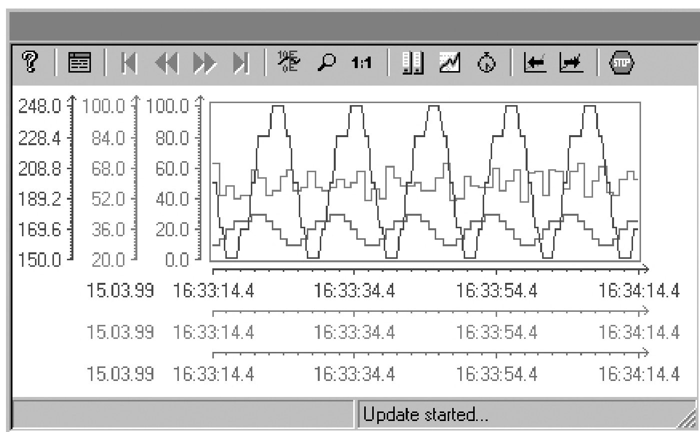



图 6-16 使用不同坐标轴显示多个趋势

（2）使用公共坐标轴显示多个趋势

如果多个趋势特征的相互比较非常重要，建议用公共坐标轴显示趋势。可为公用 X

轴分配一种颜色，如图 6-17 所示。在运行系统中，可通过缩放或坐标查询来确定精确的变量值。

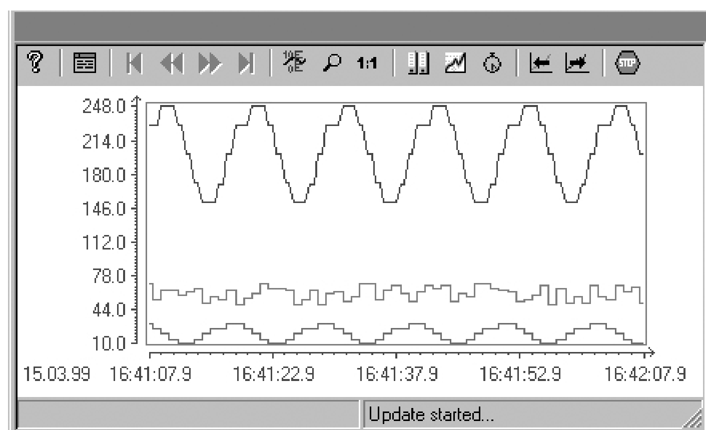



图 6-17　使用公共坐标轴显示多个趋势

（3）使用交错趋势显示多个趋势

对于范围差别较大的多个趋势值，可以选择交错趋势来显示。对于每个趋势，可以设置要显示的 Y 轴的数值范围，如图 6-18 所示。

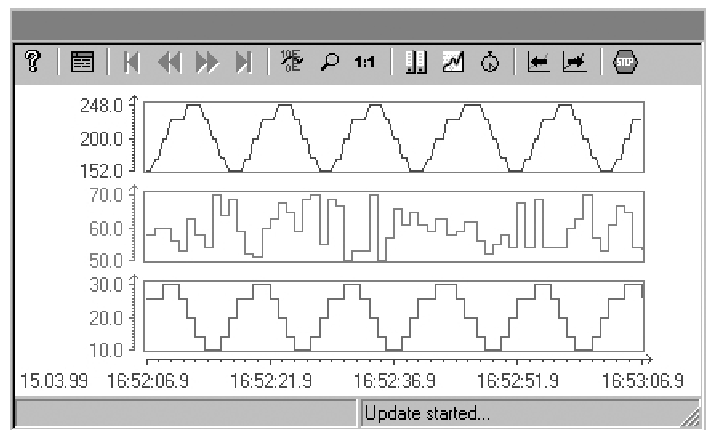



图 6-18　使用交错趋势显示多个趋势

当组态 WinCC Online Trend Control 时，对于具有共同的时间坐标轴的多个趋势而言，只有具有同一更新周期的变量才应显示在趋势窗口中。对于具有不同更新周期的变量，所有变量的时间坐标轴的长度不相同。原因是由于更新周期的不同，变量趋势在不同的时间进行更新，所以在每次改变时，时间坐标轴的末端时间可能存在细小的差别。结果是在每次改变时，所显示的趋势都将有轻微的前后跳动。

**3. 趋势显示的时间范围**

关于趋势显示的时间范围，可以通过静态显示和动态显示两种方式表示变量的变化趋势。



（1）趋势的静态显示

基于已归档数值，趋势的静态显示方式显示在所定义的时间间隔内变量的变化过程，如图 6–19 所示。

为了组态趋势的静态显示方式，可以在 WinCC Online Trend Control 的“属性”对话框的“时间轴”选项卡中取消激活“更新”选项。组态要显示的时间设置，可以通过以下方法进行，如图 6–20 所示。

- 输入开始时间和时间范围（选中“时间范围”的复选框）。
- 输入开始时间和结束时间。
- 输入开始时间和将要显示的测量点的数目（选中“测量点数量”的复选框）。

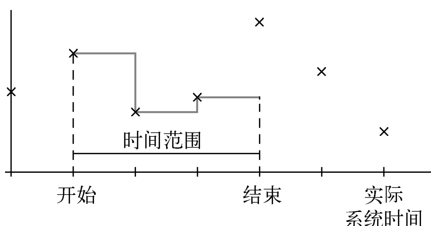



图 6–19　趋势的静态显示

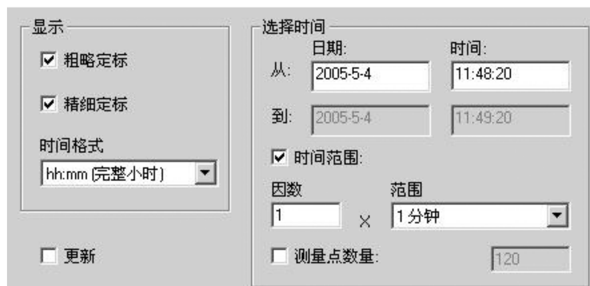



图 6–20　在“属性”对话框中组态静态显示方式

（2）趋势的动态显示

在动态显示中，趋势的结束时间始终与当前的系统时间相对应。新到达的测量值包括在显示内，如图 6–21 所示。

为了组态趋势的动态显示方式，可以在 WinCC Online Trend Control 的“属性”对话框的“时间轴”选项卡中取消激活“更新”选项。组态要显示的时间设置，可以通过以下方法进行，如图 6–22 所示。

- 输入时间范围（选中“时间范围”的复选框）。
- 输入要显示的测量点的数目（选中“测量点数量”的复选框）。
- 输入开始和结束时间之间的时间差。

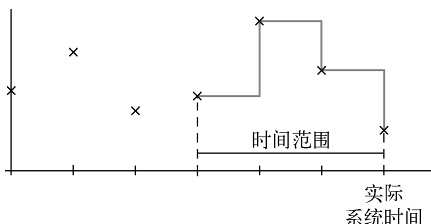



图 6–21　趋势的动态显示




图 6–22　在“属性”对话框中组态静态显示方式

**4. 组态 WinCC Online Trend Control（WinCC 在线趋势控件）**

WinCC 在线趋势控件除了可以显示归档变量外，还可以显示在线变量。

（1）在画面中插入 WinCC 在线趋势控件

1）启动图形编辑器，打开新的画面。

2）在对象选项板的“控件”选项卡上，单击“WinCC Online Trend Control”对象。

3）将鼠标指针放置在画面中想要插入控件的位置，鼠标指针变为附带对象符号的十字形光标，将控件拖至所期望的大小。

4）将打开 WinCC Online Trend Control 快速组态对话框。

5）在快速组态对话框的“常规”选项卡中组态 WinCC Online Trend Control 的属性。

6）单击“确定”按钮关闭对话框。

（2）组态趋势

在 WinCC 在线趋势控件的属性对话框中定义趋势的属性。

1）双击画面中添加的 WinCC 在线趋势控件，打开“WinCC 在线趋势控件的属性”对话框，如图 6-23 所示。




图 6-23　“WinCC 在线趋势控件的属性”对话框

2）选择“曲线”选项卡，使用 + 按钮或 - 按钮增加/减少所需趋势数目，使用 向上 按钮或 向下 按钮定义趋势顺序。

3）在“趋势”区域中选择一个趋势，在“选择归档/变量”中定义趋势的数据源：在“数据源”列表中选择在线提供数据还是从归档提供数据，通过 选择... 按钮选择相应的变量。



4）在“显示类型”列表中指定趋势的显示类型，在“线条粗细”列表中设置线条粗细。

5）在“WinCC 在线趋势控件”其他选项卡中组态趋势的其他属性。

- 在“时间轴”选项卡中，可定义访问归档值的时间范围。
- 在“数值轴”选项卡中，可定义数值范围属性，例如，缩放比例。
- 单击“确定”按钮关闭对话框，WinCC 在线趋势控件的组态即被保存。

**5. 在运行系统中操作 WinCC 在线趋势控件**

通过工具栏中的按钮可在运行系统中对趋势窗口进行操作，表 6-1 为 WinCC 在线趋势控件工具栏各个按钮的功能。

**表 6-1　在线趋势控件工具栏按钮的功能**

| 图　标 | 名　称 | 含　义 |
| --- | --- | --- |
| | 在线帮助系统 | 单击该按钮打开在线帮助，显示所有按钮的帮助信息 |
| | 打开组态对话框 | 选择该对话框以分配显示参数 |
| | 第一条数据记录 | 单击该按钮在趋势窗口中显示从第一个已归档值开始的变量趋势 |
| | 前一条数据记录 | 单击该按钮在趋势窗口中，基于当前所显示的时间间隔，显示前一时间间隔内的变量趋势 |
| | 下一条数据记录 | 单击该按钮在趋势窗口中，基于当前所显示的时间间隔，显示下一时间间隔内的变量趋势 |
| | 最后一条数据记录 | 单击该按钮在趋势窗口中，基于当前选定的时间间隔，显示以最后归档的值作为结束的变量趋势 |
| | 显示该点的数值 | 单击该按钮以确定曲线的坐标点 |
| | 放大区域 | 单击该按钮以放大趋势窗口的任意部分 |
| 1:1 | 激活原始视图 | 单击该按钮从缩放视图返回到组态的正常视图 |
| | 用于归档和变量选择的对话框 | 该按钮将打开用于归档和变量选择的对话框 |
| | 选择趋势对话框 | 单击该按钮打开用于选择可见和不可见趋势的对话框 |
| | 选择时间范围 | 单击该按钮将打开一个对话框，用于设置在趋势窗口中显示的时间范围 |
| | 前一趋势在前面 | 单击该按钮在趋势窗口的前景中显示前一个趋势 |
| | 下一趋势在前面 | 单击该按钮在趋势窗口的前景中显示下一个趋势 |
| | 停止更新 | 停止已更新的显示，再次单击此按钮将缓冲和更新值 |
| | 启动更新 | 恢复显示 |
| | 打印记录 | 单击该按钮在趋势窗口打印所显示的趋势。可在“常规”选项卡的控件属性中指定打印期间所使用的打印作业 |
| | 选择统计区域 | 单击该按钮在趋势窗口中使用垂直线来定义用于统计计算的时间段 |
| | 计算统计数据 | 单击此按钮打开统计数据窗口，其内显示指定时帧和指定趋势的最小值、最大值、平均值和标准差 |
| | 报表保存 | 将所显示趋势的控件的当前数据保存在所指示的时帧中。必须停止已更新的显示。报表将以“.csv”文件的形式保存在 WinCC 项目目录中的“导出/变量记录”下，文件名由控件的窗口标题和时间标记组成 |

（续）

| 图　标 | 名　称 | 含　义 |
|---|---|---|
| | 缩小 | 减小缩放系数 |
| | 放大 | 增加缩放系数 |
| % | 相对刻度 | 从显示绝对值切换到显示数值轴的百分比。趋势的上限和下限对应于一个 0～100% 的范围。<br>执行“相对刻度”键盘功能须满足下列先决条件：<br>• 为所有显示的趋势分配固定的值范围<br>• 趋势没有用户定标<br>• 趋势以线性缩放 |
| 右键单击趋势 | | 更新停止时，用鼠标右键单击“趋势”以显示归档名称、变量名称和相应点的坐标 |

（1）在线组态

在运行系统中，可通过 4 个按钮更改“在线表格控件”的组态：

- “打开组态对话框”按钮可以打开“WinCC 在线趋势控件的属性”对话框，定义趋势的属性。
- “用于归档和变量选择的对话框”按钮用于打开“选择归档/变量”对话框，如图 6-24 所示，可以将变量或归档链接到趋势窗口中的趋势。
- “选择趋势对话框”按钮打开用于选择可见和不可见趋势的对话框，如图 6-25 所示。




图 6-24　“选择归档/变量”对话框

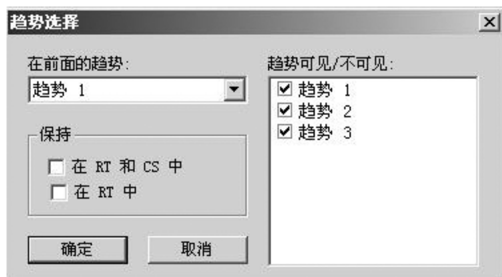



图 6-25　“趋势选择”对话框

- “选择时间范围”按钮可以打开一个对话框，如图 6-26 所示，用于指定要显示的时间范围。如果将使用公共的时间轴来显示趋势窗口中的趋势，则所指定的时间范围适用于所有趋势。

在如图 6-24～图 6-26 所示的对话框中，如果没有激活选项“在 RT 和 CS 中保持”，对设置进行的更改仅在运行系统内有效。所有已更改的设置在画面改变后是否仍然有效，将取决于“在 RT 中保持”选项。如果选项“在 RT 和 CS 中保持”已激活，则对设置进



行的所有更改也将传送给组态系统。为此，必须在图形编辑器中打开画面并对其进行重新保存。重新激活项目时，也将使用已更改的设置。



图 6–26　“选择时间范围”对话框

（2）确定点的坐标

在运行系统中，可使用趋势窗口的“显示该点的数值”按钮来确定趋势上某一点的坐标，如图 6–27 所示。为了方便确定坐标，也可对趋势线进行局部放大。

激活按钮“显示该点的数值”时，将在趋势窗口中插入一条垂直线，即作为一个标尺。在趋势窗口下方，除显示归档和变量名外，还将显示已测量值的 X 坐标和 Y 坐标。

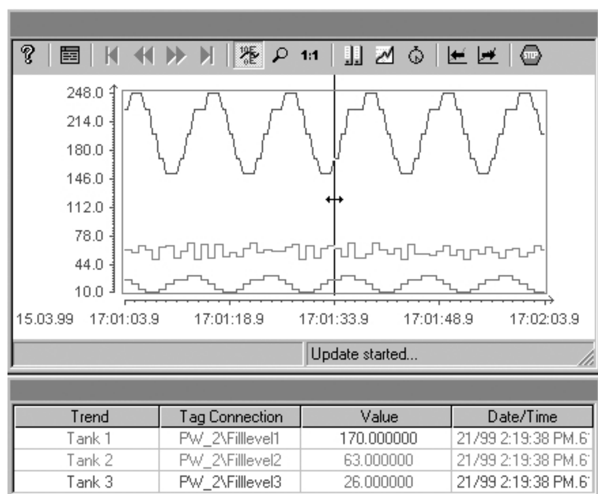


图 6–27　确定趋势上点的坐标

（3）对趋势进行局部放大

在运行系统中，可使用趋势窗口的“放大区域”按钮来对任何趋势窗口的局部进行放大。随后，用户可能会发现在放大了的显示中将更容易使用“显示该点的数值”按钮来确定某个特定测量点的坐标。使用“激活原始视图”按钮可从放大的显示切换回正常的显示模式。步骤如下：

1）单击工具栏“放大区域”按钮，显示将停止进行更新，且鼠标指针变成十字形。

2）在趋势窗口中，单击希望放大的区域的某个角。

3）按住鼠标左键拖动所要放大的区域，直到达到所期望的大小。如果高亮显示的区域包含至少两个测量值，选定的趋势区域则在趋势窗口中显示。

4）松开鼠标左键，选定区域此时便放大显示。

5）单击工具栏中的“激活原始视图”按钮，趋势窗口再次按原来所组态的正常视图进行显示。

6）单击工具栏中的“启动更新”按钮，重新启动趋势窗口中的显示更新。对于 X 轴和 Y 轴的数值范围，将默认使用预置值。

（4）生成运行系统数据的统计数据

可以在趋势窗口中生成运行系统过程数据的统计求值。对于所有已显示趋势和指定的时帧，将在统计数据窗口中显示下列结果：

- 最小值。
- 最大值。
- 平均值。
- 标准差。

生成运行系统数据的统计数据的步骤如下：

1）在趋势窗口的工具栏中，单击“选择统计区域”按钮。更新的显示将停止；过程数据继续归档。停止已更新显示，继续归档过程值。在趋势窗口的左边沿和右边沿显示两条垂直线。包含有统计数据区的窗口可显示趋势的当前上限值（OG）和下限值（UG）。

2）使用鼠标将两条垂直线拖动到期望的 x 轴位置，可指定用于计算的时帧，如图 6-28 所示。

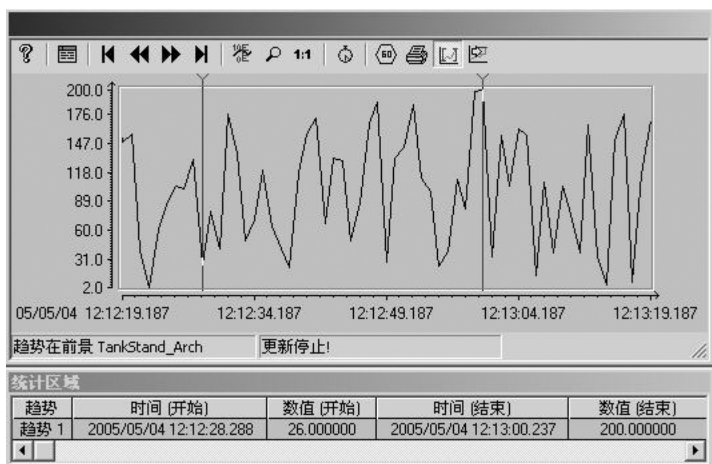



图 6-28　组态统计数据的统计区域

3）在工具栏中，单击“计算统计数据”按钮，将打开“统计”窗口，其中显示某个趋势指定时帧的计算结果，如图 6-29 所示。

统计

| 趋势 | 最小值 | 最大值 | 平均值 | 标准差 | 持续时间 | 数目 |
|---|---|---|---|---|---|---|
| 趋势 1 | 7.000000 2005/05/04 12:15:24.998 | 142.000000 2005/05/04 12:15:23.998 | 64.250000 | 56.747247 | 0 00:00:20.962 | 4 |

图 6-29　“统计”窗口显示统计数据

4）如果希望显示趋势窗口中其他趋势的结果，可在“趋势”域中选择适当的趋势。

5）为了继续显示趋势中的运行系统数据，请关闭统计数据窗口，并单击工具栏中的“启动更新”按钮。

6）如果需要对趋势窗口中未显示的过程数据进行统计分析，则可单击“选择时间范



围”按钮。在“选择时间范围”对话框中输入所期望的时帧。在指定的时帧内显示过程数据，并可计算统计数据。

## 6.3.2 以表格形式输出过程值

为了处理表格形式的数据，WinCC 以在线表格控件（WinCC Online Table Control）来提供在表格中显示过程数据的选项，可以选择显示当前值或已归档值。为了在 WinCC 在线表格控件中显示趋势，应满足下列先决条件：

- 表格只可使用归档变量。
- 每个表格可包含多达 30 000 个数值。
- 表格中的归档变量必须来源于服务器的同一个过程值归档，必须具有相同的更新周期，并且必须是在连续的周期内记录的。

### 1. 在表格中输出多个过程值

对于用表格形式显示多个过程值，有两种基本显示类型：公用的时间列和单独的时间列。

（1）公用的时间列

在公用的时间列显示类型中，表格只包含一个时间列。该列将显示第一列中变量输出的已归档时间。因此，在运行系统中，要显示三个变量值的表格将由四个列组成。

在一个带有共享时间列的表格中显示具有不同采集时间的变量会出现问题，因为变量值可能会与时间列所指示的归档时间不相符，这些数值显示为已删除。

（2）单独的时间列

在该显示类型中，每个变量的已归档时间将显示在单独的列中。因此，在运行系统中，要显示三个变量值的表格将由六个列组成。

表格形式显示多个过程值时的显示类型在 WinCC 在线表格控件的“属性”对话框的“常规”选项卡中进行组态，如图 6-30 所示，选中“公共时间列”的复选框，表格将采用公共时间列显示多个过程值，反之，则采用“单独的时间列”显示多个过程值。




图 6-30 组态表格的显示类型

### 2. 表格显示的时间范围

对于要显示的时间范围，变量的表格显示可以采用静态方式和动态方式表示变量的变化。

(1) 表格的静态显示

在静态显示中，已归档的变量值均以所定义的时间段进行显示，如图 6-31 所示。

图 6-31　表格的静态显示

在 WinCC 在线表格控件的“属性”对话框的“列”选项卡中，通过取消激活“更新”选项可组态静态显示类型，如图 6-32 所示。可通过下列方法之一来设置要显示的时间范围：

- 输入开始时间和时间范围（选中“时间范围”复选框）。
- 输入开始时间和结束时间。

图 6-32　组态表格的静态显示

(2) 表格的动态显示

在动态显示中，表格中的结束时间始终与当前的系统时间相对应，新到达的测量值将包括在显示内，如图 6-33 所示。

图 6-33　表格的动态显示



在 WinCC 在线表格控件的“属性”对话框的“列”选项卡中，通过激活“更新”选项可组态动态显示类型（参考图 6-32）。可通过下列方法之一来设置要显示的时间范围：

- 输入时间范围（选中“时间范围”复选框）。
- 指定的开始和结束时间之间的时间差。

**3. 组态 WinCC Online Table Control（WinCC 在线表格控件）**

（1）在画面中插入 WinCC 在线表格控件

1）启动图形编辑器，打开新的画面。

2）在对象选项板的“控件”选项卡中，单击“WinCC 在线表格控件”对象。

3）将鼠标指针置于要在画面中插入控件的位置。鼠标指针将变成带有附带对象符号的十字形光标，将控件拖动到所期望的大小。

4）在“WinCC 在线表格控件的属性”快速组态对话框中根据规定组态表格属性。

5）单击“确定”按钮关闭对话框。

（2）组态过程值变量的显示

1）双击在画面中添加的在线表格控件，打开“WinCC 在线表格控件的属性”对话框，如图 6-34 所示。

2）选择“列”选项卡，使用 + 按钮或 - 按钮增加/减少列的数量，使用 向上 按钮或 向下 按钮调整列的顺序。在“列”列表框中选中一个列，例如“温度”，在“选择归档/变量”中，单击 选择... 按钮，打开“选择归档/变量”对话框，为当前所选的列选择归档以及归档变量。




图 6-34 “WinCC 在线表格控件的属性”对话框

3）选择“常规”选项卡，组态在线表格控件在运行系统中的控件功能；选择“列”（右边）选项卡，组态显示过程值的时间格式和时间范围等。

4）单击“确定”按钮关闭对话框。

**4. 在线表格控件在运行系统中的操作**

在运行系统中，可以通过工具栏按钮来操作表格窗口，表格窗口的工具栏按钮多数与趋势窗口的工具栏按钮相同，这些按钮的功能请参考表 6-1。表 6-2 列出了表格窗口独有的工具栏按钮及其功能描述。

表 6-2　表格窗口工具栏按钮的功能

| 图标 | 名　　称 | 含　　义 |
|---|---|---|
|  | 编辑 | 一旦激活此按钮，就可双击任何表格单元来编辑其内容，这将停止已更新的显示 |
|  | 前一列在前面 | 该按钮允许将表格窗口中当前的最后列作为表格的第一个数据列来显示。在多项选择中，在时间列中显示相应的已归档时间。在多项选择中，在时间列中显示相应的已归档时间 |
|  | 下一列在前面 | 允许将表格窗口当前的最后列作为表格的第一个数据列来显示。在多项选择中，在时间列中显示相应的已归档时间。在多项选择中，在时间列中显示相应的已归档时间 |

（1）在运行系统中编辑表格单元

使用“编辑”按钮，可以将实验值或修正值插入到过程值归档中。单击该按钮后，将暂停表格显示，此时双击表格内任何单元都可以改变此单元的内容。

可在静态显示中查看编辑过程期间生成的数值。通过取消激活“更新”选项，在“WinCC 在线表格控件的属性”对话框的“列”选项卡中可对静态显示进行组态。如果通过编辑改变了多项选择的时间标志，则只在数据库的第一列中写入新数值。其他列的时间标志将保持不变。

再次单击“编辑”按钮，将返回到运行系统模式中的表格显示。

（2）生成运行系统数据的统计数据

1）在表格窗口的工具栏中，单击“选择统计区域”按钮。如果希望计算所有表格值的统计数据，可单击“停止更新”按钮来停止已更新显示，继续归档过程数据。

2）为了指定计算的时间段，可使用鼠标来选择所期望时帧的表格行，如图 6-35 所示。对于具有不同时帧的不同数据列，可选择不同的时间范围，用于统计数据的计算。

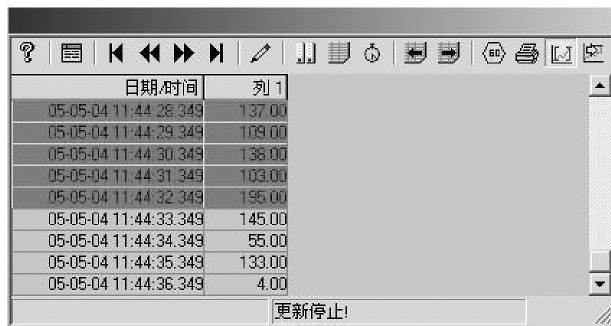



图 6-35　组态统计数据的时间段

3）在工具栏中，单击“计算统计数据”按钮，将打开“统计”窗口，其中显示某个趋势指定时帧的计算结果，如图 6-36 所示。

统计

| 列 | 最小值 | 最大值 | 平均值 | 标准差 | 持续时间 | 数目 | 时间 (开始) | 时间 |
|---|---|---|---|---|---|---|---|---|
| 列 1 | 28.00 05-05-04 11:46:21.198 | 181.00 05-05-04 11:46:17.19 | 101.50 | 60.52 | 00:00:05.00 | 6 | -05-04 11:46:16.1 | -04 1 |

图 6-36　“统计”窗口显示统计数据

4）如果希望显示其他表格列的结果，则可在“列”域中选择适当的列。

5）为了继续显示表格中的运行系统数据，必须关闭统计数据窗口，并单击工具栏中的“启动更新”按钮。



6）如果需要对表格窗口中未显示的过程数据进行统计分析，可单击“选择时间范围”按钮，在“选择时间范围”对话框中输入所期望的时帧，将在指定的时帧内显示过程数据，并可计算统计数据。

## 6.4　应用实例

### 6.4.1　组态要求

**1. 归档基本参数**

● 归档名：Mesured_values_Station_1_10。

● 归档类型：过程值归档。

● 归档变量：Trend_1——采集周期 1 秒，归档周期 1 秒。

　　IVar_Meas——采集周期 2 分钟，归档周期 2 分钟。

**2. 归档数据库参数**

● 快速归档：1 天 300 MB。

● 慢速归档：7 天 280 MB。

● 选择当前的日期和时间作为数据段改变时间。

**3. 趋势图属性**

● 趋势 1：名称→Trend_1，颜色→红色，变量→Trend_1，类型→线性连接各点。

● 趋势 2：名称→IVar_Meas，颜色→绿色，变量→IVar_Meas，类型→线性连接各点。

**4. 趋势窗口属性**

● 趋势 1：X 轴：坐标→时间，时间范围→1 分钟，更新→是。

　　Y 轴：坐标→温度，粗略定标→25，精细定标→5，开始和结束→0－100。

● 趋势 2：X 轴：坐标→时间，时间范围→1 分钟，更新→是。

　　Y 轴：坐标→压力，近似比例→20，精确比例→5，开始和结束→0－120。

### 6.4.2　组态方法与步骤

根据上述组态要求，组态变量记录系统的方法与步骤如下：

1）在 WinCC 变量管理中组态类型为“无符号 16 位数”的内部变量 Trend_1 和 Ivar_Meas，用做变量记录系统的归档变量。

2）在 WinCC 项目管理器的浏览窗口双击“变量记录”，或用鼠标右键单击“变量记录”，然后在弹出的菜单中选择“打开”，打开 WinCC 变量记录。

3）在 WinCC 变量记录的浏览窗口用鼠标右键单击“定时器”，然后在弹出的菜单中选择“新建”，打开“定时器属性”对话框，如图 6-37 所示，在“名称”框中输入“2 分钟”，在“基准”列表框中选择“1 分钟”，在“系数”框中输入数字 2，单击“确定”按钮，将新增 1 个时间为 2 分钟的定时器。

4）在 WinCC 变量记录的浏览窗口用鼠标右键单击“归档”，然后在弹出的菜单中选择“归档向导”，打开“创建归档”对话框。单击“下一步”按钮，在打开的“创建归档：步骤－1－”对话框中，在“归档名称”的文本框中输入归档名称，例如“Process-




ValueArchive”，选择归档类型为“过程值归档”，单击“下一步”按钮。在打开的“创建归档：步骤 -2-”对话框中，单击 选择... 按钮，选择归档的变量 Trend_1 和 Ivar_Meas，如图 6-38 所示。单击“完成”按钮，结束归档向导。




图 6-37　组态新的定时器




图 6-38　组态归档变量

5）在 WinCC 变量记录的数据窗口选中刚刚组态的归档“ProcessValueArchive”，在表格窗口用鼠标右键单该归档的归档变量，例如“Trend_1”，然后在弹出的菜单中选择“属性”，打开“过程变量”属性对话框，如图 6-39 所示，选择“归档”选项卡，按照组态要求设置该归档变量的归档参数。按同样的方法组态归档变量“Ivar_Meas”的归档参数，如图 6-40 所示。




图 6-39　归档变量“Trend_1”的归档参数






图 6-40　归档变量“Ivar_Meas”的归档参数

6）在变量记录的浏览窗口选择“归档组态”，在数据窗口用鼠标右键单击“TagLogging Fast”，然后在弹出的菜单中选择“属性”，打开“TagLogging Fast”对话框，根据组态要求组态快速归档的参数，如图 6-41 所示。用同样的方法组态慢速归档的参数，如图 6-42 所示。




图 6-41　快速归档的参数

7）在变量记录中的工具栏单击“保存”按钮或单击菜单“文件”“保存”菜单项保存所做的组态。

8）在图形编辑器中新建 1 个过程画面，选择对象选项板的“控件”→“WinCC Online Trend Control”添加到画面并调整到合适大小。

9）双击画面中的 WinCC 在线趋势控件，打开“WinCC 在线趋势控件的属性”对话框。




图 6-42　慢速归档的参数

10）选择“曲线”选项卡，如图 6-43 所示，单击 + 按钮，在“趋势”列表中增加 1 个趋势。在“趋势”列表中选择 1 个趋势，在“名称”框中输入“Trend_1”。在“显示”组中选中“可见”前的复选框，并单击 颜色... 按钮，打开调色板将曲线的颜色设置




图 6-43　组态趋势“Trend_1”的属性



为红颜色。在“选择归档/变量”组中设置“数据源”为“归档变量”，单击 选择... 按钮选择归档变量“Trend_1”。在“显示类型”下拉列表中选择“线性连接点”。

11）选择“时间轴”选项卡，根据组态要求组态趋势“Trend_1”的 X 轴（时间轴）的属性：在“标签”文本框中输入“时间”。选中“更新”前的复选框。选中“时间范围”前的复选框，并在“系数”框中输入数字 1，在“区域”下拉列表框中选择“1 分钟”。

12）选择“数值轴”选项卡，根据组态要求组态趋势“Trend_1”的 Y 轴（数值轴）的属性：在“标签”文本框中输入“温度”。在“比例”组中，在下拉列表框中选择“线性”，选中“粗略定标”并在其后的文本框中输入数字 25，选中“精细定标”并在其后的文本框中输入数字 5。在“格式化”组中，在“小数位”和“标尺小数位”后的文本框中分别输入数字 0。在“范围选择”组中，取消选中“自动”前的复选框，并在“从”后的文本框中输入数字 0，在“到”后的文本框中输入数字 100。

13）按照步骤 10）、11）、12）的方法根据组态要求组态趋势“Ivar_Meas”的属性。

14）单击“确定”按钮退出“WinCC 在线趋势控件的属性”对话框，在图形编辑器中保存所做的组态。

15）在 WinCC 项目管理器的浏览窗口选择“计算机”，在数据窗口用鼠标右键单击当前组态消息系统的计算机名称，在弹出的菜单中选择“属性”命令，打开“计算机属性”对话框。选择“启动”选项卡，在“WinCC 运行时的启动顺序”列表框中选中“变量记录运行系统”的复选框。单击“确定”按钮退出“计算机属性”对话框。运行 WinCC 项目。

16）为了测试组态的变量记录系统和趋势窗口，打开 WinCC 变量模拟器（相关内容参考本书第 4 章 4.7.1 节），添加仿真变量 Ivar_Meas 和 Trend_1，选择仿真函数并设置相关参数后，单击“Start Simulation”按钮开始仿真，如图 6-44 所示。趋势窗口的仿真运行结果如图 6-45 所示。

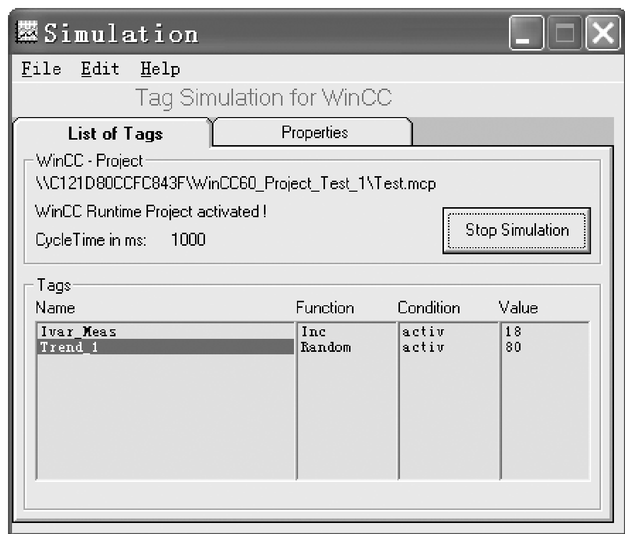



图 6-44　WinCC 变量模拟器模拟变量 Ivar_Meas 和 Trend_1

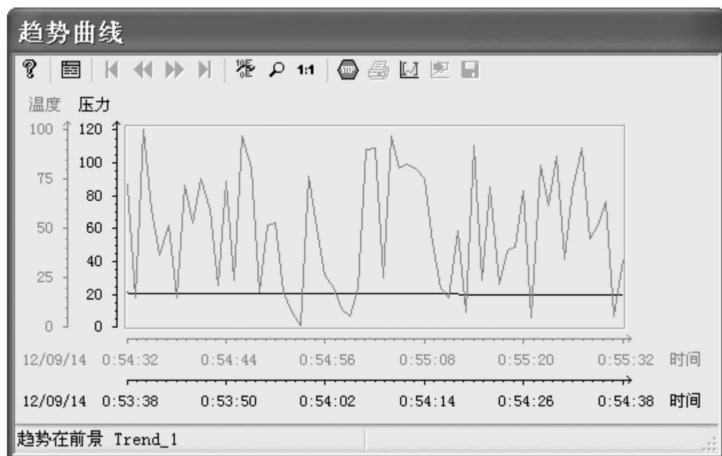



图6-45　趋势窗口的仿真运行结果

## 6.5　组态技巧与技术问答

本节将介绍一些WinCC组态变量记录系统的技巧，并针对组态过程中遇到的一些常见的技术问题进行分析和解答。

### 6.5.1　WinCC如何界定快速归档与慢速归档?

WinCC默认将1分钟作为划分快速归档和慢速归档的界限，即归档周期低于1分钟的周期连续归档称为快速归档，而归档周期大于1分钟的周期连续归档称为慢速归档。

从WinCC V6.0版本开始，快速归档和慢速归档的归档周期界限可以由用户自行设定，该参数可以在快速归档属性的“归档内容”选项卡中设置，详细设置方法和步骤如下：

1）在WinCC变量记录的浏览窗口选中“归档组态”，在数据窗口用鼠标右键单击“TagLogging Fast”，在弹出的菜单中选择“属性”，打开设置快速归档属性的“TagLogging Fast”对话框。

2）在对话框中选择“归档内容”选项卡，如图6-46所示，选中“周期小于等于以

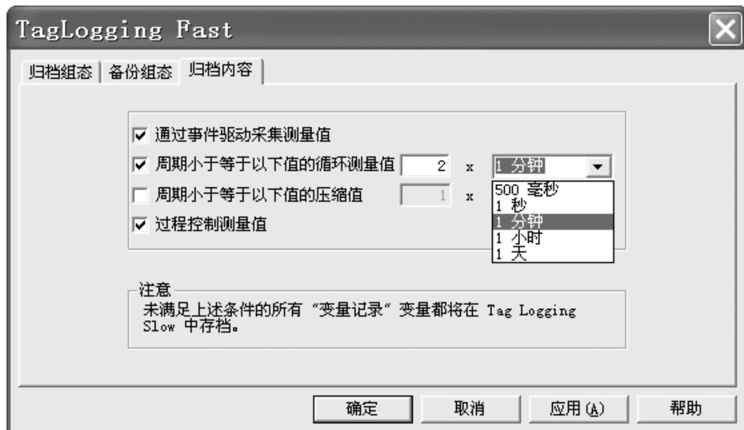



图6-46　组态快速归档和慢速归档的界限



下值的循环测量值”前的复选框，并在其后的下拉列表框中设定归档周期的基数，例如“1 分钟”，在文本框中设定归档周期的系数，例如“2”，则新的划分快速归档和慢速归档的界限为 2 分钟。

3）单击“确定”按钮保存设置并退出对话框。

除了设置过程值归档的快速归档和慢速归档界限，还可以设置压缩值归档的快速归档和慢速归档界限，如图 6-46 所示，选中“周期小于等于以下值的压缩值”前的复选框，并在后面的文本框中设置归档周期的计数和系数。

## 6.5.2 归档数据库尺寸的计算方法

前文已述及，过程值的归档可以对工业现场的历史生产数据做一个完备的记录，从而对当前和今后的工作起到很强的借鉴作用。对于过程值的归档来说，很重要的一个问题是归档数据库尺寸的计算。

在计算归档数据库尺寸时，除了要考虑工业现场的实际情况（例如数据量的大小）外，还需要先考虑以下三个问题：

- 首先，一条慢速变量归档记录与一条快速变量归档记录占用的硬盘空间是不同的，因此要计算数据库尺寸就得分两种情况考虑。
- 第二，WinCC V6.0 版本中的快、慢速归档的归档周期界限可以由用户自行设定（参考本章 6.5.1 节的内容）。
- 第三，数据归档类型分为四类，只有周期连续归档的数据才能定量地计算其占用的数据库尺寸，因此当设定时间段和设置数据库尺寸大小时，需要考虑其他数据归档类型的数据，留出相应的余量。

为了详细解释数据库尺寸的计算方法，这里举例加以说明：某用户希望进行两个月的数据归档，其中有 5000 个变量的归档周期是 2 分钟，50 个变量的归档周期是 2 秒钟，单个的数据片段时间设置为 1 周，怎样设置相应的数据段尺寸？

归档数据库尺寸的计算方法如下（以 1 分钟划分快速归档和慢速归档的界限）：

- 慢速归档时一条变量归档记录占用 32 B 的空间，每个变量以 2 分钟为归档周期，一周之内会产生 5 040 条记录，若有 5 000 个变量的归档，则单个数据片段的大小计算为：32 × 5 000 × 5 040 = 806 400 000 B，约等于 800 MB，考虑到留出 20% 的余量，设定单个数据片段为 1 GB 所有数据归档期限是两个月，因此所有段的尺寸为单个片段尺寸乘以单个片段的个数，即：1 GB × 9 = 9 GB。
- 快速归档时一条变量归档记录占用 3 B 的空间，每个变量以 2 秒钟为归档周期，一周之内会产生 302 400 条记录，若有 50 个变量的归档，则单个数据片段的大小计算为：3 × 50 × 302 400 = 45 360 000 B，约等于 46 MB，考虑到留出 20% 的余量，设定单个数据片段为 60 MB，所有数据归档期限是两个月，因此所有段的尺寸为单个片段尺寸乘以单个片段的个数，即：60 MB × 9 = 540 MB。

## 6.5.3 合理设置过程值归档的采集周期和归档周期的方法

采集周期确定读取过程变量过程值的时间间隔，而归档周期确定何时将过程值保存到归档数据库中，归档周期总是采集周期的整数倍。在组态过程值归档时，必须合理地

设置采集周期和归档周期，否则将增加系统的负载。因为采集周期设置后，对于 WinCC 后台运行系统来说是永久负载（通信），例如，

- 设置采集周期：5 秒，归档/显示周期：5 秒，该设置可以优化通信负载。
- 设置采集周期：1 秒，归档/显示周期：5 秒，该设置同样可以优化通信负载。
- 设置采集周期：1 秒，归档/显示周期：1 秒，该设置使得归档数据量很大，给系统带来不必要的通信负载。
- 设置采集周期：1 秒，归档/显示周期：10 秒，该设置使得归档时可以采用一定的算法来处理过程值，例如取过程值的平均值进行归档。

另外，归档周期的起始点设置合理可以延迟归档以及更均匀地分配归档负载，例如：

- 在三个周期中归档过程值：每分钟、每 2 分钟以及每 3 分钟。这使得每 6 分钟就出现一次高归档负载。给这三个周期中的每个周期分配一个不同的起始点，例如，在每分钟的第 15 秒归档、在每 2 分钟的第 30 秒归档、在每 3 分钟的第 45 秒归档。这可分散归档负载。
- 很多过程值必须每 10 秒归档一次，为了分散归档负载，例如，可以用不同的起始点组态两个“10 秒”周期，在第 0 秒以及第 5 秒时执行归档。

## 6.5.4　整点启动过程值归档的组态方法

归档周期确定何时将过程值保存到归档数据库中，归档周期的起始点取决于 WinCC 运行系统的启动时间或所使用时间的起始点。在工业现场，为了统计上的方便，往往需要在整点启动过程值的归档。在 WinCC 中，可以通过两种方法实现整点启动归档：

- 在 WinCC 变量管理的 System Info 通道下创建变量用于获取系统的小时值，该变量的值随着系统时钟而变化的，一旦改变就在全局动作中置位归档控制变量来启动归档。
- 通过脚本实现整点启动归档。

### 1. 通过 System Info 通道的变量实现整点启动归档

在 WinCC 的系统信息（System Info）通道的通信驱动程序下创建的 WinCC 变量，可以专门用于记录系统信息，相关的详细内容请参考本书第 4 章 4.7.2 节。通过 System Info 通道的变量实现整点启动归档的方法和详细步骤如下：

1）用鼠标右键单击 WinCC 项目管理器浏览窗口的“变量管理”，在弹出的菜单中选择“添加新的驱动程序”，打开“添加新的驱动程序”对话框，选择“System Info.chn”，单击“打开”按钮，在“变量管理”中增加“系统信息”项。

2）用鼠标右键单击“变量管理”中的“系统信息”项，在弹出的菜单中选择“新驱动程序的连接”，打开“连接属性”对话框。在“名称”后的文本框中输入连接名称（例如“SystemInfo”），单击“确定”按钮，创建一个系统信息通道的连接“SystemInfo”。

3）用鼠标右键单击新建的连接“SystemInfo”，在弹出的菜单中选择“新建变量”，打开“变量属性”对话框，在“名称”框中输入变量名称 TagHour，“数据类型”下拉列表框中选择“无符号 16 位数”，单击“地址”框后的“选择”按钮，打开“系统信息”对话框，设置该变量的格式，如图 6-47 所示，在“函数”后的下拉列表框中选择“小时”函数，单击“确定”按钮返回到“变量属性”对话框，根据需要设置变量的其他属



性后单击“确定”按钮，退出“变量属性”对话框。




图 6-47　创建系统信息变量 TagHour

4）在 WinCC 变量管理中创建“二进制变量”类型的内部变量 stararchive，用做归档控制变量。

5）在 WinCC 项目管理器的浏览窗口单击“全局脚本”前的“+”，用鼠标右键单击“C－Editor”，在弹出的菜单中选择“打开”，打开全局脚本 C 编辑器。在全局脚本 C 编辑器的左边窗口用鼠标右键单击“动作”，在弹出的菜单中选择“新建”，在打开的“动作”窗口中输入以下代码：

```
int gscAction( void)
{
    #pragma option( mbcs)
    if( GetTagBit( "startarchive" ) ==0)
        SetTagBit( "startarchive" ,1) ;
    return 0;
}
```

6）在全局脚本 C 编辑器的工具栏中，单击“信息/触发”按钮，打开“属性”对话框，如图 6-48 所示。单击“添加”按钮，打开“添加触发器”对话框，在“事件”后的下拉列表框中选择“变量”，单击“变量名”后的按钮 ... ，打开“变量－项目”选择框，选择变量 TagHour，并将周期设置为“有变化时”。单击“确定”按钮退出“添加触发器”对话框和“属性”对话框。

7）在全局脚本 C 编辑器保存组态的全局动作，名称为“startarchive. pas”。

8）打开 WinCC 变量记录，在变量记录的浏览窗口选择“归档”，在数据窗口选择需要组态整点启动的过程值归档，在表格窗口用鼠标右键单击该过程值归档的归档变量，在弹出的菜单中选择“属性”，打开“过程变量属性”对话框，如图 6-49 所示。选择“归档”选项卡，在“归档类型”中选择“周期”，在“动作”组内设置“开始:”的变量为 startarchive。单击“确定”按钮退出对话框。





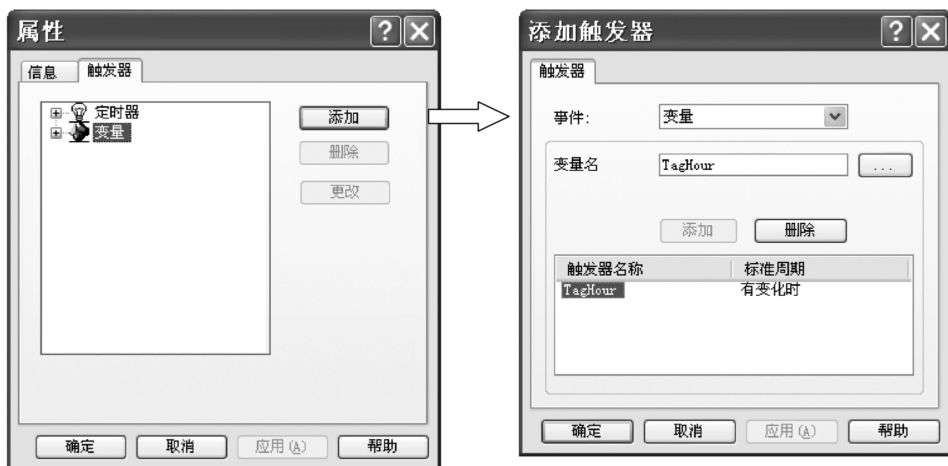



图 6-48　组态全局动作的触发器




图 6-49　设置过程变量归档属性

9）在 WinCC 变量记录中保存所做的组态。

**2. 通过脚本实现整点启动归档**

通过脚本实现整点启动归档需要在全局脚本 C 编辑器中创建一个项目函数，然后在 WinCC 变量记录中设置归档变量的归档类型为周期的选择性过程值归档，并将归档启动事件设置为在全局脚本中创建的项目函数，具体组态方法和步骤如下：

1）在 WinCC 变量管理中创建“二进制变量”类型的内部变量 stararchive，用做归档控制变量。

2）在 WinCC 项目管理器的浏览窗口单击“全局脚本”前的“+”，用鼠标右键单击“C－Editor”，在弹出的菜单中选择“打开”，打开全局脚本 C 编辑器。在全局脚本 C 编辑器的左边窗口用鼠标右键单击“项目函数”，在弹出的菜单中选择“新建”，在打开的“项目函数”窗口中输入以下代码：



```
BOOL cycliarchive( )
{
    #pragma code( " kernel. dll" ) ;
        void GetLocalTime( SYSTEMTIME * lpsz) ;
    #pragma code( ) ;
    SYSTEMTIME time;
    int t1 ;
    GetLocalTime( &time) ;
    t1 = time. wMinute;
    if( t1 = = 00)
  {
    SetTagBit( " startarchive" ,1) ;
  return( BOOL) ( GetTagBit( " startarchive" ) ) ;
  }
}
```

3）在全局脚本 C 编辑器保存组态的项目函数，名称为“cycliarchive”。

4）打开 WinCC 变量记录，在变量记录的浏览窗口选择“归档”，在数据窗口选择需要组态整点启动的过程值归档，在表格窗口鼠标右键单击该过程值归档的归档变量，在弹出的菜单中选择“属性”，打开“过程变量属性”对话框，如图 6-50 所示。选择“归档”选项卡，在“归档类型”中选择“周期”，在“动作”组内单击“开始:”的 C 脚本后的按钮 ...，打开“函数浏览器”，选择项目函数 cycliarchive。单击“确定”按钮退出对话框。




图 6-50 设置过程变量归档属性

5）在 WinCC 变量记录中保存所做的组态。

# 第7章

# 报 表 系 统

报表用于归档过程数据和完整的生产周期，以创建班次报表、输出批量数据，或者对生产制造过程进行归档以用于验收测试等。WinCC 报表系统允许创建和输出项目文档报表和运行系统数据报表。

本章将详细介绍 WinCC 创建和输出报表的方法。

## 7.1 WinCC 报表系统简介

WinCC 报表系统的报表分为以下两种：

- 项目文档报表，用于输出 WinCC 项目的组态数据。
- 运行系统数据报表，用于在项目运行期间将过程数据输出到日志中。

### 1. 项目文档报表

WinCC 可创建和输出的项目文档包括：

- WinCC 项目管理器。
- 图形编辑器。
- 报警记录。
- 变量记录。
- 全局脚本。
- 文本库。
- 用户管理器。
- 用户归档。
- 时间同步。
- 警报器编辑器。
- 画面树管理器。
- 设备状态监控。
- OS 项目编辑器。

项目文档报表由存在项目数据库表中的数据组成，这些数据对应的是各 WinCC 编辑器（例如图形编辑器、变量记录等）包括管理器（例如项目管理器）中的组态设置。项目数据库可在项目文件夹中找到，其文件名为“项目名.db”，是一个只读文件，不能对其删除或重命名。一个新项目的所有项目文档报表都已事先组态好，并能在各编辑器的



组态方式下的“文件”菜单中进行打印。

2. 运行系统数据报表

表 7-1 为 WinCC 报表系统可以创建和输出的运行记录数据文档类型。

表 7-1 WinCC 报表系统可记录的数据文档类型

| 记录系统 | 报表对象 |
|---|---|
| 报警记录系统 | 消息顺序报表 |
| | 消息报表 |
| | 归档报表 |
| 变量记录系统 | 变量记录表格 |
| | 变量趋势 |
| 用户归档运行系统 | 用户归档表 |
| CSV 文件 | CSV 数据源表 |
| | CSV 数据趋势 |
| 通过 ODBC 记录数据 | ODBC 数据库域 |
| | ODBC 数据库表 |
| 自身 COM 服务器 | COM 服务器 |
| 硬拷贝输出 | 硬拷贝 |

运行系统数据报表由项目运行数据库表组成，包括报警和变量存档档案库。运行数据库可在项目文件夹中找到，其文件名为“项目 RT. db”。运行系统数据报表只有在生成报表的编辑器数据处于运行状态时才能打印。

各种报表的结构和组态几乎是一样的，不同的是报表的布局、打印输出、启动过程中数据以及与动态对象的链接。

## 7.2 WinCC 报表编辑器

WinCC 报表编辑器是 WinCC 基本软件包的一部分，提供了报表的创建和输出功能。创建是指创建报表布局，可以分为页面布局和行布局。输出是指打印输出报表，在打印作业中可以找到时序表、输出介质和输出定义的范围等。

### 7.2.1 页面布局

在页面布局中，报表编辑器为可视化结构提供了静态、动态和系统对象。

每个新的 WinCC 项目都有若干个预先组态好的页面布局可供选择，可以在报表编辑区中打开进行编辑修改。用户也可以通过报表编辑器生成自己的布局。

1. 在页面布局中设置报表

页面布局在几何上分割为多个不同的区域。页面范围对应于布局的整个区域，可为该区域定义打印页边距。正确的操作是：首先为页眉、页脚或公司标志组态可打印区域的页边距，然后才对应于报表数据输出的其余可打印区域进行组态。在可打印区域内定义的该区域称为“页面主体”，如图 7-1 所示。

报表布局包括静态层和动态层。静态层包括布局的页眉和页脚，用于输出公司名称、公司标志、项目名称、布局名称、页码、时间等。动态层包括输出组态和运行系统数据的动态对象。

只有静态对象和系统对象可插入到静态层，静态对象和动态对象均可插入动态层。

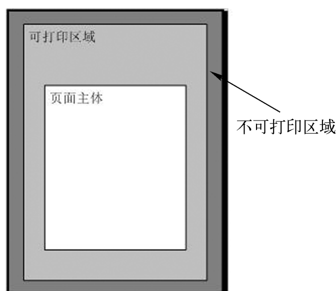


图 7-1　页面布局

如果需要，插入到页面布局动态部分中的对象可进行动态扩展。例如，当动态表中的对象被提供数据时，可扩展该表以允许输出表中的所有数据。如果在布局的动态部分中还存在其他对象，则对其进行相应的移动。因此，具有固定位置的对象必须插入到布局的静态部分。

每个页面布局由 3 个页面组成：

（1）封面

封面是页面布局的固定组件。因此，可以为各个报表设计一个单独的封面。

（2）报表内容

在页面布局的该部分中，定义了报表输出时的结构和内容。可以使用系统对象来定义报表内容。报表内容具有静态组件和动态组件（组态层）。如果必要，则报表内容的动态部分在输出时将分散为各种不同的后续页面，因为直到输出时才能知道存在多少数据。

（3）封底

封底是页面布局的固定部分。因此，可以为每个报表设计一个单独的封底。

**2. 页面布局编辑器**

页面布局编辑器作为报表编辑器的组件，用于创建和动态化报表输出的页面布局。页面布局编辑器仅能用于在 WinCC 项目管理器中打开的当前项目。所保存的布局作为该项目的基准。

在 WinCC 项目管理器浏览窗口中，选中“报表编辑器”，则其下方出现两个子目录：布局和打印作业。双击“布局”或用鼠标右键单击“布局”，然后在弹出的菜单中选择“打开页面布局编辑器”，将打开页面布局编辑器，如图 7-2 所示。

页面布局编辑器是根据 Windows 标准构建的，它具有工作区、工具栏、菜单栏、状态栏和各种不同的选项板。打开页面布局编辑器后，将出现带默认设置的工作环境，用户可根据喜好排列选项板和工具栏或隐藏它们。

页面布局编辑器包括以下组成部分：

（1）工作区

页面的可打印区将显示为灰色区，而页体部分将显示为白色区。工作区中的每个画面都代表一个布局，并将保存为独立的 rpl 文件。布局可按照 Windows 标准进行扩大和缩小。

（2）菜单栏

菜单栏始终可见。不同菜单上的功能是否激活，取决于状况。

（3）工具栏

工具栏提供一些按钮，以便快速地执行页面布局编辑器常用命令。根据需要，可在



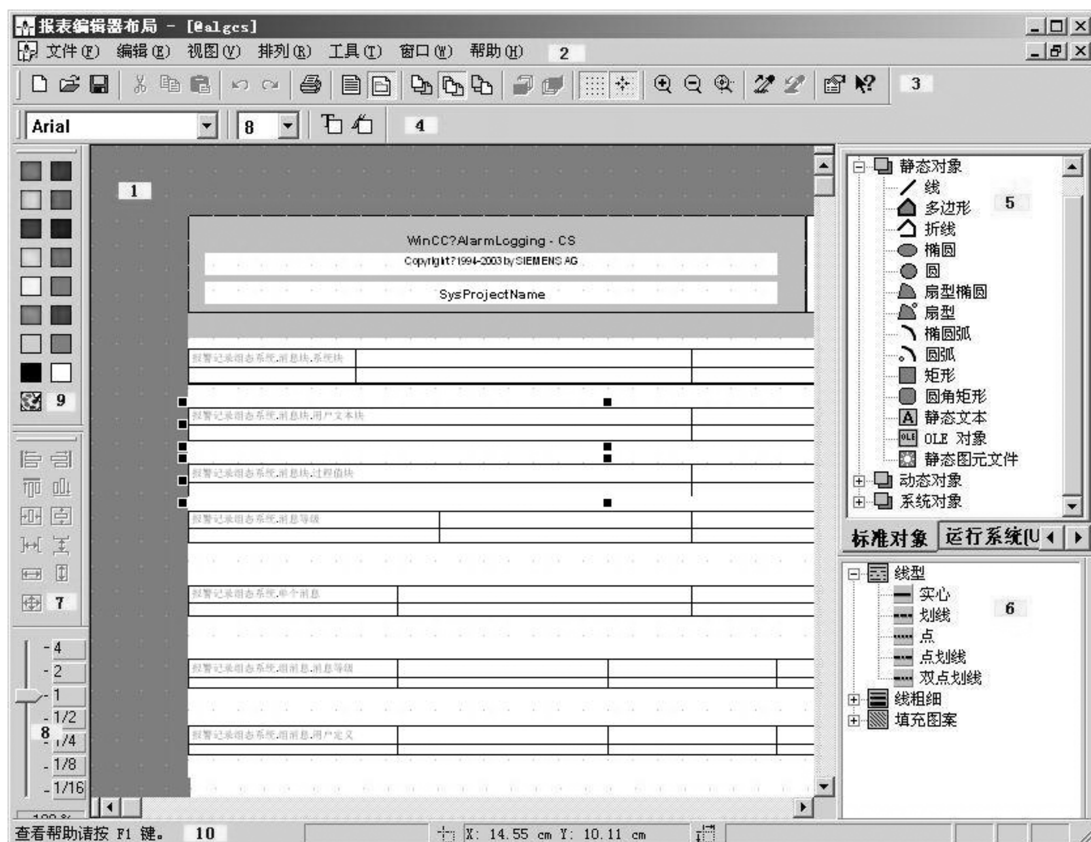



图 7-2　页面布局编辑器

屏幕的任何地方隐藏或移动工具栏。

（4）字体选项板

字体选项板用于改变文本对象的字体、大小和颜色，以及标准对象的线条颜色。

（5）对象选项板

对象选项板包含标准对象、运行系统文档对象、COM 服务器对象和项目文档对象，这些对象用于构建布局。

（6）样式选项板

样式选项板用于改变所选对象的外观。根据对象的不同，可改变线段类型、线条粗细或填充图案。

（7）对齐选项板

使用对齐选项板可改变一个或多个对象的绝对位置以及改变所选对象之间的相对位置，并可对多个对象的高度和宽度进行标准化。

（8）缩放选项板

缩放选项板提供了用于放大或缩小活动布局中对象的两个选项：使用带有标准缩放因子的按钮或使用滚动条。

（9）调色板

调色板用于为选择的对象涂色。除了 16 种标准颜色之外，还可定义自己的颜色。

（10）状态栏

状态栏位于屏幕的下边沿，可根据需要将其隐藏。其中，它显示提示、所选对象的位置信息以及键盘设置。

**3. 构建页面布局的对象**

构建页面布局的对象包含在页面布局编辑器对象选项板中，包括标准对象、运行系统文档对象、COM 服务器对象和项目文档对象。

（1）标准对象

标准对象由静态对象、动态对象和系统对象组成。其中，静态对象用于构建可视化页面布局。页面布局的静态部分和动态部分中都可插入静态对象。

动态对象可与具有当前对象有效数据格式的数据源相连接，该数据可按 WinCC 布局输出，见表 7-2。只能将动态对象插入到页面布局的动态部分中。

**表 7-2　页面布局的动态对象**

| 图标 | 对　　象 | 含　　义 |
|---|---|---|
| | 嵌入布局 | 项目文档的布局可与“所嵌入的布局”动态对象嵌套。对象只用于 WinCC 已建布局中的项目文档 |
| | 硬拷贝 | 使用“硬拷贝”对象类型，可将当前屏幕和内容的画面或其所定义部分输出到日志中，也可输出当前所选择的画面窗口 |
| | ODBC 数据库域 | 使用“ODBC 数据库域”对象类型，可通过 ODBC 接口将来自某些数据源的文本输出到日志中 |
| | ODBC 数据库表 | 使用“ODBC 数据库表”对象类型，可通过 ODBC 接口将来自某些数据源的表格输出到日志中 |
| | Tag（变量） | 输出具有“变量”对象类型的运行系统中的“变量”值。当然，只有在项目被激活时才能输出变量值。在运行系统中，也可调用脚本进行输出 |

系统对象用做系统时间、报表的当前页码以及项目和布局名称的占位符。系统对象只能插入到页面布局的静态部分中，见表 7-3。

**表 7-3　页面布局的系统对象**

| 图标 | 对　　象 | 含　　义 |
|---|---|---|
| | 日期/时间 | 使用“日期/时间”系统对象，可将输出的日期和时间占位符插入到页面布局中。在打印期间，系统日期和时间均由计算机进行添加 |
| | 页码 | 使用“页码”系统对象，可将报表或日志当前页码的占位符插入页面布局 |
| | 项目名称 | 使用“项目名称”系统对象，可将项目名称占位符插入页面布局 |
| | 布局名称 | 使用“布局名称”系统对象，可将布局名称占位符插入页面布局 |

（2）运行系统文档对象

运行系统文档对象用于输出运行系统数据的日志，见表 7-4。输出选项可使用“对象属性”对话框进行组态。日志的数据可在输出时从已链接的数据源中提取。只能将运行系统文档对象插入页面布局的动态部分中。



表 7-4　页面布局的运行系统文档对象

| 对　　象 | 含　　义 |
| --- | --- |
| WinCC 报警控件表格 | “WinCC 报警控件/表格”对象用于以表格的格式输出消息列表 |
| WinCC 功能趋势控件画面 | “WinCC 功能趋势控件/画面”对象用于以趋势形式输出作为来自过程值、压缩的和用户归档的其他变量函数的过程数据 |
| WinCC 在线表格控件表格 | “WinCC 在线表格控件/表格”对象用于以表格格式输出来自相关的变量记录归档的过程数据 |
| WinCC 在线趋势控件画面 | “WinCC 在线趋势控件/画面”对象用于以趋势形式输出来自相关的变量记录归档的过程数据 |
| WinCC 用户归档控件表格 | “WinCC 用户归档控件/表格”对象用于以表格格式输出来自用户归档的数据或视图 |
| 报警记录 RT 归档日志 | “归档报表”对象链接到消息系统，并将保存在消息归档中的消息输出到表格中 |
| 报警记录 RT 消息日志 | “消息报表”对象链接到消息系统，并将消息列表中的当前消息输出到表格中 |
| 用户归档运行系统表 | “用户归档运行系统表”对象链接到用户归档，并将用户归档和视图中的运行系统数据输出到表格中 |
| CSV 供应商表格 | “CSV 数据源表”对象链接到 CSV 文件。文件中所包含的数据均输出到表中。数据必须具有预先定义的结构 |
| CSV 供应商趋势 | “CSV 数据源趋势”对象链接到 CSV 文件。文件中所包含的数据均可输出到曲线中。数据必须具有预先定义的结构 |

（3）COM 服务器对象

为了使用 COM 服务器对象，必须将 COM 服务器项目器集成到 WinCC 中。该 COM 服务器使对象可用于记录数据。采用这种方式，可以将用户指定的数据集成到 WinCC 日志中。COM 服务器对象的形式和属性均由 COM 服务器记录器确定。使用 COM 服务器记录器传递 COM 服务器对象的描述。用于选择输出数据的选项均由当前 COM 服务器对象确定。只能将 COM 服务器对象插入页面布局的动态部分中。

（4）项目文档对象

项目文档对象包括图形编辑器中的动作、报警记录 CS、全局脚本等。项目文档对象可用于所组态数据的报表输出。只能将项目文档对象插入页面布局的动态部分。

项目文档对象将与 WinCC 组件严格链接，对象类型是固定的。根据要输出的组态数据的类型和大小，使用了“静态文本”、“动态图元文件”或“动态表”对象类型。

对于某些使用了“动态图元文件”和“动态表”对象类型的对象，可改变用于输出的组态数据的选择。

## 7.2.2　行布局

### 1. 在行布局中设置报表

行布局包括静态层和动态层。静态层包括页眉和页脚，可以以纯文本的形式输出公司名称、项目名称、布局名称等。动态层包括用于输出报警记录消息的动态表。

每个行布局由3个区域组成：

（1）页眉

页眉是行布局的固定组件，将随每一页面输出。行布局中的页眉最多可由10行组成，不能插入图形。

（2）日志内容（表）

在行布局的该部分中，定义了日志输出时的结构和内容。报警记录选择选项以及定义日志内容的过滤标准均可用于报警输出。

（3）页脚

页脚是行布局的固定组件，将随每一页面输出。行布局的页脚最多可由10行组成。不能插入图形。

**2. 行布局编辑器**

行布局编辑器仅用于创建和编辑消息顺序报表的行布局，每个行布局包含一个连接到 WinCC 消息系统的动态表。附加的对象不能添加到行布局。可在页眉和页脚中输入文本。

在 WinCC 项目管理器浏览窗口中，选中“报表编辑器”，则其下方出现两个子目录：布局和打印作业。用鼠标右键单击“布局”，然后在弹出的菜单中选择“打开行布局编辑器”，将打开行布局编辑器，如图7-3所示。




图7-3　行布局编辑器

行布局编辑器包括以下组成部分：

（1）菜单栏

菜单栏始终可见。不同菜单上的功能是否激活，取决于状况。

（2）工具栏

工具栏在行布局编辑器中始终可见。工具栏上有不同的按钮，可以快速激活菜单命令的功能。按钮是否激活，取决于状况。



（3）页眉区域

页眉区域允许输入文本以创建行布局的页眉。

（4）表格区域

用于输出的表格的设计在表格区域中显示。所组态的列标题和列宽（每列字符数）将显示。使用该区域中的按钮，可组态表格用于输出。

（5）页脚区域

页脚区域用于输入文本以创建行布局的页脚。

（6）页面大小区域

页面大小区域用于设置行布局的行数和列数。

（7）页边距区域

页边距区域用于设置行布局输出的页边距。

（8）状态栏

可在屏幕下端找到状态栏，它包含了有关工具栏按钮、菜单命令以及键盘设置的提示。

## 7.2.3 打印作业

WinCC 中的打印作业对于项目和运行系统文档的输出极为重要。在布局中组态输出的外观和数据源。在打印作业中组态输出介质、打印数量、开始打印的时间以及其他输出参数。

每个布局必须与打印作业相关联，以便进行输出。WinCC 中提供了各种不同的打印作业，用于项目文档。这些系统打印作业均已与相应的 WinCC 应用程序相关联，既不能将其删除，也不能对其重新命名。

可在 WinCC 项目管理器中创建新的打印作业，以便输出新的页面布局。WinCC 为输出行布局提供了特殊的打印作业，行布局只能使用该打印作业输出，不能为行布局创建新的打印作业。

**1. 创建新的打印作业**

在 WinCC 项目管理器浏览窗口中，选中“报表编辑器”，则其下方出现两个子目录：布局和打印作业。用鼠标右键单击“打印作业”，然后在弹出的菜单中选择“新建打印作业”，则自动建立一个名称为“PrintJob001”的打印作业。每创建一个新的打印作业，打印作业名称中的数字就将递增一次。在创建打印作业后，现有的所有打印作业均显示在项目管理器的数据窗口中。

**2. 组态打印作业的属性**

在 WinCC 项目管理器浏览窗口中，选中“打印作业”，在数据窗口双击需要修改属性的打印作业或用鼠标右键单击，然后在弹出的菜单中选择“属性”，将打开“打印作业属性”对话框，如图 7-4 所示。

在“打印作业属性”对话框的“常规”选项卡中输入名称、所使用的布局以及启动参数。此外，还可对调用打印作业时确定发生哪些事件的附加选项进行设置。

（1）打印作业的名称

打印作业的名称将显示在“名称”区域中，可在此重命名打印作业。随产品提供的








图 7-4 “打印作业属性”对话框

系统打印作业不能重命名，因为它们与 WinCC 的不同应用程序直接相关联。

（2）用于输出的布局

通过“布局”可选择所期望的输出布局。此时，如果打印作业是“@ 报表报警记录 RT 消息序列”，则只能选择行布局。只有在该作业中，才能选择“行式打印机的行布局”选项的复选框。如果选择复选框，则消息顺序报表将输出到本地行式打印机。如果没有选择该复选框，则消息顺序报表将按页面格式输出到可选打印机。

所提供的系统打印作业以及在其中设置的布局均可用于项目文档的输出。因此，系统打印作业应不与其他布局相关联，否则项目文档将不再正确运行。

（3）打印作业列表标记

图形编辑器包括一个属于打印作业列表的应用程序窗口。如果 WinCC 画面中集成了该打印作业列表，则可显示打印作业，用于在运行期间进行记录，并启动输出。在打印作业列表中，可对打印作业的显示进行设置。可选择下列视图：

- 所有打印作业。
- 仅系统打印作业。
- 仅用户自定义打印作业。
- 具有所选“标记打印作业列表”选项的打印作业。

“标记打印作业列表”选项允许在运行期间将所需的打印作业选择放在一起。

（4）显示参数对话框

为了使运行系统文档更灵活，许多记录参数都已经进行了动态化。这将允许在运行期间改变记录输出。从对话框中选择“显示参数对话框”选项，当运行期间调用打印作业时，将调用“参数”对话框。可在运行期间修改输出参数。该对话框也将允许选择或改变用于输出的打印机。

（5）调用打印机安装选项

为在页面布局中输出日志，可在运行期间改变用于输出的打印机。为此，可在对话



框中选择“打印机设置”选项，当运行期间调用打印作业时，将调用用于选择打印机的对话框。

（6）设置启动参数

在“启动参数”区中，可设置启动时间和输出周期，该设置主要用于定期输出运行系统文档中的日志（例如，换班报表）。项目文档不需要启动参数，因为项目文档不是周期性输出。对于已组态了启动参数或周期性调用的打印作业，可在 WinCC 项目管理器中根据打印作业列表的不同符号来识别。

### 3. 选择打印范围

在“打印作业属性”对话框的“选择”选项卡中可指定将要打印的范围，可指定页面范围的选择或将要输出的数据的时间范围，如图 7-5 所示。如果使用消息顺序记录打印输出在线数据，则“选择”选项卡上所有的设置均取消激活。

（1）选择页面范围

在“页面范围”区域中可指定输出时将要打印多少，既可输出单个页面，也可输出页面范围或所有的页面。

（2）选择数据的时间范围

可使用“相对”选项来指定用于输出的相对时间范围（从打印启动时间开始），下列时间间隔都可用于相对时间范围：所有、年、月、星期、日和小时。“绝对”选项将允许为输出的数据指定绝对的时间范围。

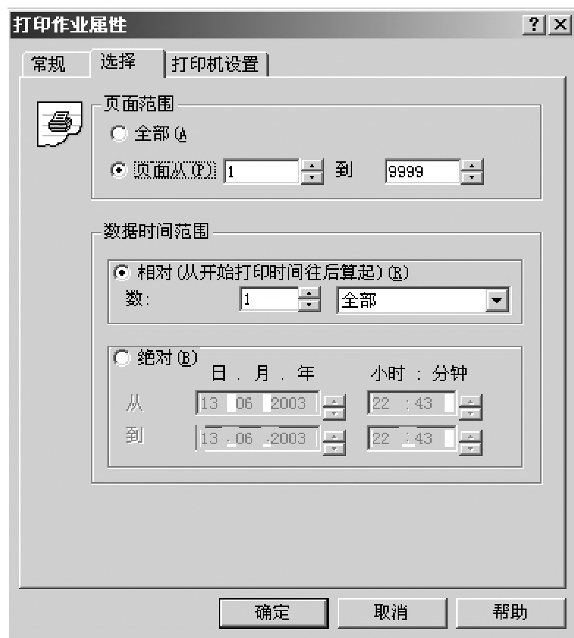



图 7-5　设置打印作业的打印范围

需要说明的是，如果在打印作业的布局中组态了时间范围，则该设置将比打印作业设置的时间范围优先级高。时间范围的选择只与报警记录和变量记录中的运行系统文档有关。

如果为输出数据的选择设置了过滤标准，则除了“打印作业属性”对话框中的设置以外，这些过滤标准均应予以考虑，唯一例外的是，如果使用了“DATETIME”过滤标准，则忽略打印作业中的时间范围设置。

如果通过报警控件中的按钮启动了“@ ReportAlarmLoggingRT…”打印作业，则将忽略布局和打印作业的设置，因为输出数据的选择是从报警控件传送过来的。

**4. 指定打印机**

在“打印作业属性”对话框的“打印机设置”选项卡上，可指定用于输出的一台或多台打印机。使用“打印机优先级”区域中的列表可指定打印机的使用次序，也可在此处指定打印缓冲区的设置以及输出到文件的设置，如图 7-6 所示。




图 7-6　指定打印机

（1）打印输出到

在“打印机设置”选项卡中，可指定用于输出的一台或多台打印机，打印机将按其优先级次序排列。报表和日志均输出到“1. )”所设置的打印机。如果该打印机出现故障，则将自动输出到“2. )”所设置的打印机。对于第三台打印机也采用相同的操作步骤。如果查找不到可以运行的任何打印机，则打印数据将保存到硬盘上的某个文件中，这些文件均存储在项目目录的“PRT_OUT”文件夹中。一旦打印机发生故障，操作系统将输出一条出错消息，此时，可以有下列选择：

- 忽略出错消息（建议使用）：当打印机再次可用时，自动打印未决的消息（打印作业仍然位于假脱机程序中）。
- 重复：如果按下“重复”按钮，那么操作系统将尝试重新输出仍然位于假脱机程序中的打印作业，只有在打印机再次准备就绪时才值得这样操作。
- 取消：如果按下“取消”按钮，将删除引起出错的打印作业，打印数据也因而丢失。于是操作系统将试图打印假脱机程序中的下一个打印作业。



强烈建议通过单击“忽略”按钮对操作系统的消息进行确认，如果单击“取消”按钮，所有打印已失败且正在等待的打印作业都会丢失。

如果故障影响了用于消息顺序报表的行式打印机，则单击“取消”可禁止打印机，此时，可能必须重新启动 WinCC 项目才能再次启用打印机。检测打印机是否发生故障所需的时间取决于所组态的打印机连接的超时设置，通过操作系统中的打印机连接属性可对超时进行设置。

也可组态两台替代打印机用于基于行的消息顺序报表，行式打印机必须连接并安装在执行记录的计算机上，如上述方式进行切换。如果没有找到已准备就绪的任何打印机，则除了出现操作系统信息外，将显示 WinCC 对话框，该对话框将提供有关打印作业状态的信息，并输出打印机故障，可在该对话框中关闭消息顺序报表。如果在该对话框中关闭消息顺序报表，则一旦所组态的打印机重新准备就绪，消息顺序报表将立即自动再次打开。

（2）输出到文件

可将报表和日志输出到文件。如果打印行布局中的消息顺序记录，输出为文件的功能会取消激活。

（3）设置磁盘空间的最小值

为输出日志，必须有用于输出到打印机的缓冲区和用于输出到文件的缓冲区。在“打印机设置”选项卡中，可为磁盘的空闲空间组态两个限制值，用于每个缓冲区。如果与限制值发生冲突，将触发下列动作：

- 产生警告：如果正在使用的硬盘的空闲磁盘空间低于此处的设置值，则将在“WinCC_SStart_01. log”文件中创建一个条目。如果运行系统激活，则生成一条系统消息，该消息指出了将要面临的资源瓶颈。
- 放弃记录的触发：如果正在使用的硬盘的空闲磁盘空间低于此处的设置值，将拒绝触发日志。此外，还将在“WinCC_SStart_01. log”文件中创建一个条目。如果运行系统激活，将生成一条系统消息，以引起对丢失日志的注意。

在设置磁盘空间的最小值时，有以下建议和限制条件：

- 打印机的超时设定应保持在较低水平（大约 10 s）。
- 每个 LPT 端口上只能安装一台打印机。
- 当计算机用于基于行的消息顺序报表时，必须在消息顺序报表的打印作业中组态作替换用的打印机。
- 不支持混合使用基于行和页面的消息顺序报表，作替换用的打印机也必须为行式打印机。

## 7.3 组态报表

在 WinCC 中生成和打印报表是一个简单的过程。在每个 WinCC 项目生成的默认布局和打印作业，通常可以满足大多数报表的需要，一般不需要用户设计布局。如果希望生成一个新的报表或修改一个已有的报表，基本步骤如下：

1）在布局编辑器生成或修改报表的布局。如果使用系统已有的布局，则此步骤可以

省略。可以修改已有布局或生成新的布局。

2）生成或修改一个打印作业。打印作业定义报表的打印时间表、打印范围和打印机的选择等。如果使用已有的打印作业，则此步可以省略。

3）组态报表，在计算机属性中激活“报表运行系统”。

## 7.3.1 组态行式打印机上的消息顺序报表

消息顺序报表允许按时间顺序输出在项目中产生的所有消息的列表，通常在行式打印机上输出，消息一旦到达，打印机将自动打印。使用行式打印机输出时，必须将该打印机与执行记录的计算机进行本地连接，必须选择打印作业中的“行式打印机的行布局”复选框。

### 1. 报表输出的先决条件

报表输出的先决条件有两条：

- 将要输出消息顺序报表的行式打印机必须从本地连接到执行记录的计算机上。
- 必须在执行记录的计算机的启动列表中激活消息顺序报表，当 WinCC 项目激活时，消息顺序报表将自动启动。

激活消息顺序报表的方法：在 WinCC 项目管理器的浏览窗口中选中“计算机”，在数据窗口用鼠标右键单击计算机名，在弹出的菜单中选择“属性”命令，打开“计算机属性”对话框。选择“启动”选项卡，选中“WinCC 运行时的启动顺序”列表中“消息顺序报表”前的复选框。单击“确定”按钮，退出“计算机属性”对话框。

### 2. 创建行布局

创建行布局的步骤如下：

1）单击 WinCC 项目管理器浏览窗口中的“报表编辑器”条目，显示“布局”和“打印作业”条目。选择“布局”条目，现有的所有布局将显示在数据窗口中。

2）双击名称为“@ CCAlgRtSequence. rp1”的行布局，该布局将显示在行布局编辑器中。

3）在“页面大小”区，指定每个页面的行数和列数（每行字符数），在“页边距”区，指定用于页边距宽度的字符数。

4）编辑页眉和页脚的内容，这些内容将在每页上输出。

5）单击“选择”按钮，打开“报警记录运行系统：报表－表格列选择”对话框，如图 7-7 所示，可以在对话框中指定输出的数据。当关闭对话框时，所选择的列及其宽度在表格区域以每行字符数显示。如果每行的字符数太多，将显示相应的消息。

6）保存对行布局所作的修改，并关闭行布局编辑器。

### 3. 组态打印作业

单击 WinCC 项目管理器浏览窗口中的“报表编辑器”条目，显示“布局”和“打印作业”条目。选择“打印作业”条目，现有的所有打印作业显示在数据窗口中。双击打印作业“@ Report Alarm Logging RT Message Sequence”，打开其属性对话框。在对话框的“常规”选项卡中，选中“行式打印机布局”前的复选框，并指定布局“@ CCAlgRtSequence. rp1”，如图 7-8 所示。在“打印机设置”选项卡中设置期望的打印机。单击“确定”按钮，退出对话框。



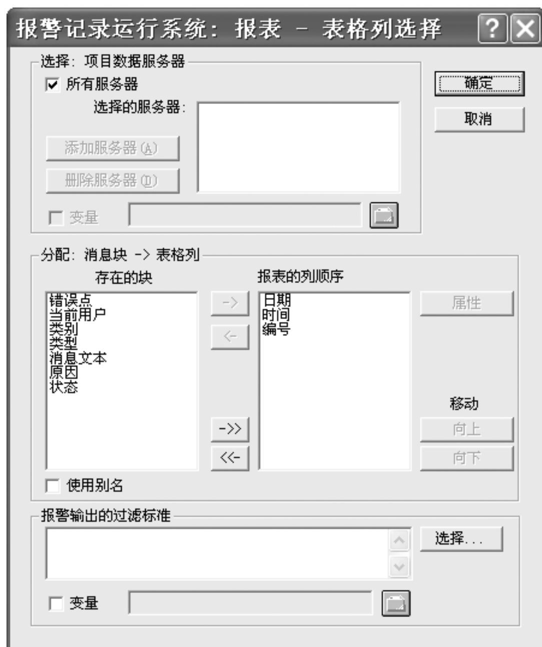



图 7-7 “报警记录运行系统:报表 - 表格列选择”对话框

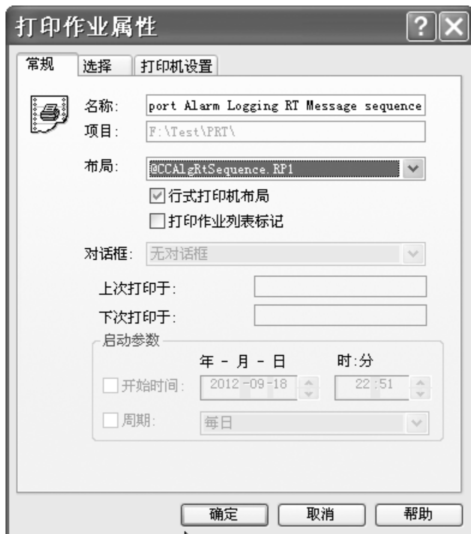



图 7-8 组态消息顺序报表的打印作业

## 7.3.2 组态页面布局的消息顺序报表

消息顺序报表也可在页面布局中输出，此时可以由直接与本地执行记录的计算机连接的打印机输出，也可以由网络打印机输出。以页面布局输出时，如果进入的消息填满一个页面或按下操作按钮启动打印输出页面，将执行打印输出作业。

WinCC 报表编辑器中包含了预先组态好的页面布局和打印作业，用于输出基于页面布局的消息顺序报表。其中，页面布局名称为“@ CCAlgRtOnlineMessages. rpl”，对应的打印作业为“@ Report Alarm Logging RT OnlineMessages”。用户也可以选择自己创建页面布局和打印作业。

### 1. 创建页面布局

创建消息顺序报表的页面布局的步骤如下：

1）单击 WinCC 项目管理器浏览窗口中的“报表编辑器”条目，显示“布局”和“打印作业”条目。用鼠标右键单击“布局”，在弹出的菜单中选择“新建页面布局”，则自动创建一个名称为“NewRPL0. RPL”的布局。用鼠标右键单击新建的布局，在弹出的菜单中选择“重命名页面布局”，在打开的对话框中输入布局名称“MessageSequenceReport. RPL”。

2）双击新创建的布局，打开页面布局编辑器。单击工具栏上的图标，将布局定位到“报表内容”部分。单击工具栏上的图标，让当前的页面布局显示在动态层。

3）从对象选项板中选择“运行系统”→“报警记录运行系统”→“消息报表”对象插入到页面布局的动态部分，并拖放到合适的尺寸，如图 7-9 所示。

4）双击页面布局中的“消息报表”对象，打开“对象属性”对话框，选择“连接”选项卡，如图 7-10 所示。

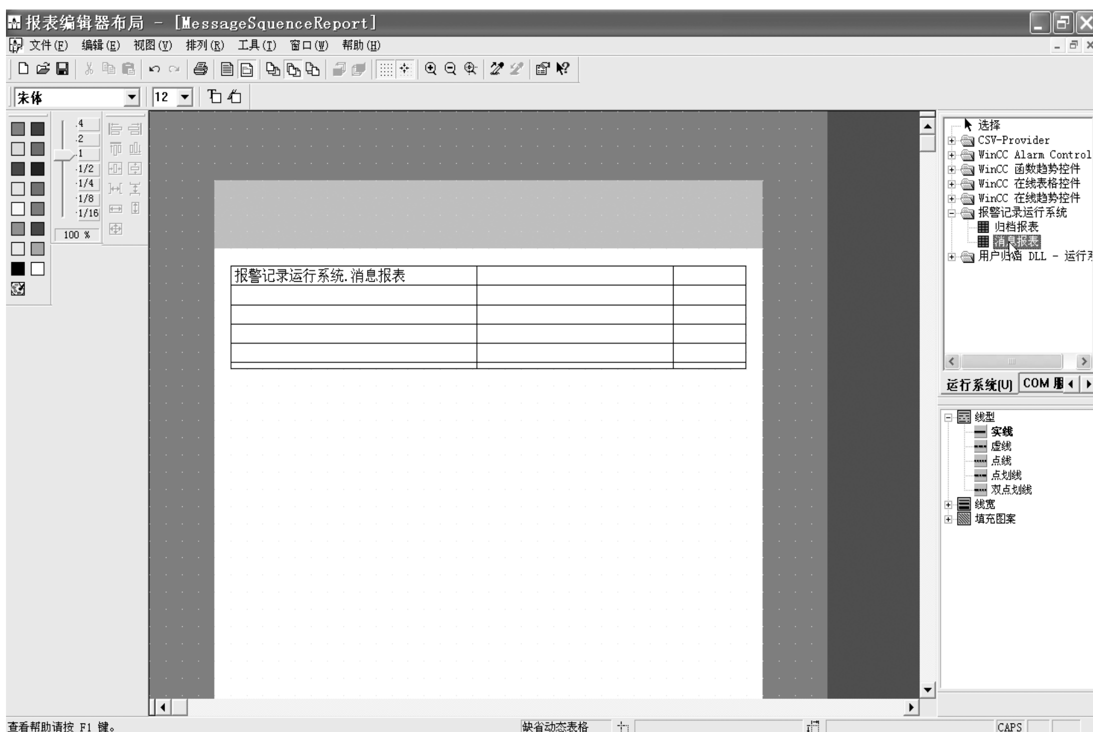



图 7–9　在页面布局中插入“消息报表”对象




图 7–10　消息报表的“对象属性”对话框

5）双击“选择”或选中“选择”项并单击“编辑”按钮，打开“报警记录运行系统：报表－表格列选择”对话框，如图 7–11 所示。将“存在的块”列表中需要在消息报表中打印的消息块选中，然后单击按钮→将其移至“报表的列顺序”列表中。在“报表的列顺序”列表中选中消息块，然后单击“属性”按钮，可以编辑消息块的属性。

6）双击图 7–10 中的“时间基准”或选中“时间基准”，然后单击“编辑”按钮，打开“报警记录运行时间：时间基准输入”对话框，可以设置消息报表输出消息的时间基准，例如“应用项目设置”。



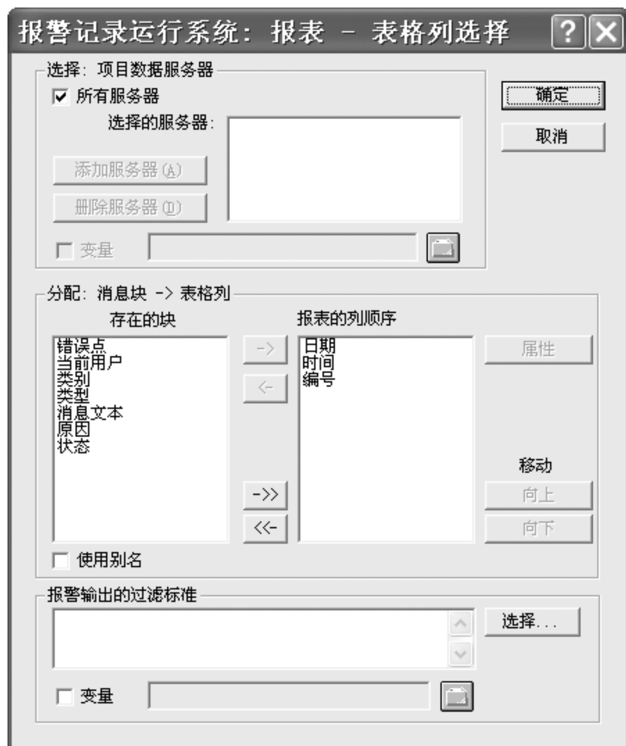



图 7-11　“报警记录运行系统：报表 - 表格列选择”对话框

7）在图 7-9 的工作区的空白位置单击鼠标右键，然后在弹出的菜单中选择“属性”，打开布局的“对象属性”对话框，如图 7-12 所示，选择“属性”→“几何”，可以设置输出的报表的纸张大小等属性。




图 7-12　布局的“对象属性”对话框

8）在页面布局编辑器的工具栏中，单击“保存”按钮保存组态的页面布局。

**2. 组态打印作业**

组态打印作业的步骤如下：

1）单击 WinCC 项目管理器浏览窗口中的“报表编辑器”条目，显示“布局”和

“打印作业”条目。用鼠标右键单击“打印作业”，在弹出的菜单中选择“新建打印作业”，则自动创建一个打印作业。

2）在项目管理器浏览窗口选中“打印作业”，在数据窗口右键单击新建的打印作业，在弹出的菜单中选择“属性”，打开“打印作业属性”对话框。

3）选择“常规”选项卡，在“名称”框中输入打印作业名称“PrintMessageSquenceReport”，在布局后的下拉列表框中选择“MessageSquenceReport. RPL”布局。

4）选择“打印机设置”选项卡，设置期望的打印机。

5）单击“确定”按钮，退出对话框，完成组态。




图 7-13　组态打印作业属性

注意：必须在“计算机属性”对话框的“启动”选项卡中激活“报表运行系统”。

## 7.3.3　组态归档消息报表

消息顺序报表输出的是运行系统的在线消息，而消息归档报表可以输出来自于消息归档的消息。报表编辑器中没有预先组态用于输出归档消息的报表布局和打印作业，因此需要用户组态页面布局和对应的打印作业。

**1. 创建页面布局**

1）单击 WinCC 项目管理器浏览窗口中的“报表编辑器”条目，显示“布局”和“打印作业”条目。用鼠标右键单击“布局”，在弹出的菜单中选择“新建页面布局”，则自动创建一个布局。用鼠标右键单击新建的布局，在弹出的菜单中选择“重命名页面布局”，在打开的对话框中输入布局名称“Alarm_Log_Archive. RPL”。

2）双击新创建的布局，打开页面布局编辑器。单击工具栏上的图标，将布局定位到“报表内容”部分。单击工具栏上的图标，让当前的页面布局显示在动态层。



3）从对象选项板中选择“运行系统”→“报警记录运行系统”→“归档报表”对象插入到页面布局的动态部分，并拖放到合适的尺寸（与图 7-9 相似，此处不再插图）。

4）双击页面布局中的“消息报表”对象，打开“对象属性”对话框，选择“连接”选项卡（参考图 7-10）。

5）双击“选择”或选中“选择”项，然后单击“编辑”按钮，打开“报警记录运行系统：报表 - 表格列选择”对话框。将“存在的块”列表中需要在消息报表中打印的消息块选中，然后单击按钮 ->，将其移至“报表的列顺序”列表中。在“报表的列顺序”列表中选中消息块，然后单击“属性”按钮，可以编辑消息块的属性。

6）双击“时间基准”或选中“时间基准”，然后单击“编辑”按钮，打开“报警记录运行时间：时间基准输入”对话框，可以设置消息报表输出消息的时间基准，例如“应用项目设置”。

7）在页面布局编辑器工作区的空白位置单击鼠标右键，然后在弹出的菜单中选择“属性”，打开布局的“对象属性”对话框，选择“属性”→“几何”，可以设置输出的报表的纸张大小等属性。

8）在页面布局编辑器的工具栏中，单击“保存”按钮，保存组态的页面布局。

**2. 组态打印作业**

组态打印作业的步骤如下：

1）单击 WinCC 项目管理器浏览窗口中的“报表编辑器”条目，显示“布局”和“打印作业”条目。用鼠标右键单击“打印作业”，在弹出的菜单中选择“新建打印作业”，则自动创建一个打印作业。

2）在项目管理器浏览窗口选中“打印作业”，在数据窗口右键单击新建的打印作业，在弹出的菜单中选择“属性”，打开“打印作业属性”对话框。

3）选择“常规”选项卡，在“名称”框中输入打印作业名称“Print Alarm_Log_Archive”，在布局后的下拉列表框中选择“Alarm_Log_Archive. RPL”布局。

4）选择“打印机设置”选项卡，设置期望的打印机。

5）单击“确定”按钮，退出对话框，完成组态。

注意：必须在“计算机属性”对话框的“启动”选项卡中激活“报表运行系统”。

### 7.3.4 组态变量记录报表

WinCC 报表系统允许对来自变量记录的过程值和来自用户归档的数据进行记录，下列选项可用：

- 以表格形式输出来自过程值归档和压缩归档的变量数据。
- 以趋势形式输出来自过程值归档和压缩归档的变量数据。
- 以趋势形式输出作为来自过程值、压缩的或用户归档的其他变量函数的过程值。
- 以表格形式输出来自用户归档的数据

本节以表格形式输出来自过程值归档的变量数据为例，介绍组态变量记录报表的方法。

**1. 组态页面布局**

WinCC 报表编辑器预先组态了用于输出表格形式变量记录报表的页面布局，布局名




称为“@CCTlgRtTables.RPL”，用户可在此页面布局的基础上进行组态。

在 WinCC 项目管理器的浏览窗口选择“报表编辑器”，将显示“布局”和“打印作业”条目。选择“布局”条目，则所有预先组态的布局都在数据窗口显示。双击名称为“@CCTlgRtTables.RPL”的布局，打开页面布局编辑器，如图 7-14 所示。可以看到在布局的工作区添加了“变量表格”对象。

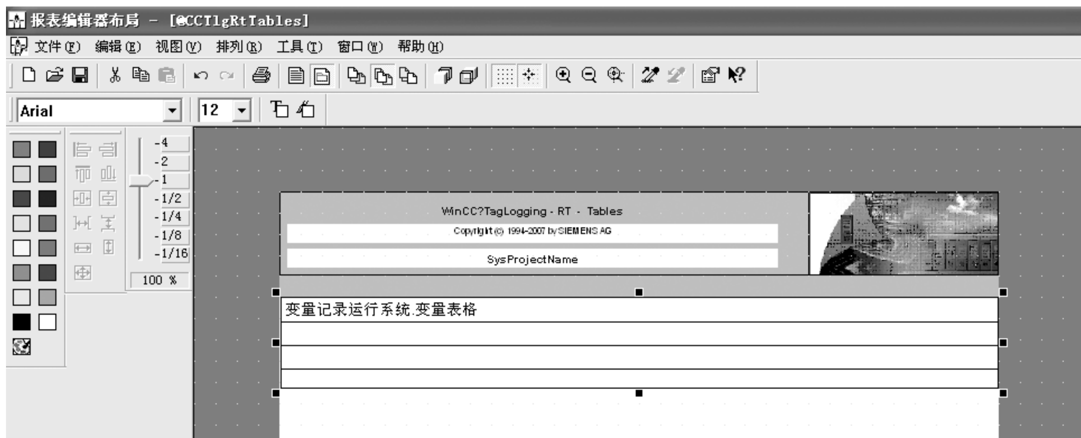



图 7-14　“@CCTlgRtTables.RPL”布局

双击“变量表格”对象，打开“对象属性”对话框，选择“连接”选项卡，如图 7-15 所示。选中“变量选择”后，单击“编辑”按钮，打开“变量记录运行系统：报表的变量选择”对话框，单击“添加”按钮，添加归档变量，如图 7-16 所示。




图 7-15　变量表格的“对象属性”对话框




图 7-16　添加报表输出的归档变量



需要对添加的归档变量组态输出时的格式，在“变量记录运行系统：报表的变量选择”对话框的“当前的选择和顺序”列表中选中一个归档变量，单击“属性”按钮，打开变量的属性对话框，如图7-17所示。选择“数值”选项卡可以组态变量的输出格式，选择“时间”选项卡可以组态时间列的格式。




图 7-17　归档变量的“属性”对话框

在页面布局编辑器工作区的空白位置单击鼠标右键，在弹出的菜单中选择“属性”，打开布局的“对象属性”对话框，选择“属性”→“几何”，可以设置输出的报表的纸张大小等属性。

所有组态完成后，单击布局编辑器工具栏的“保存”按钮，保存对页面布局的组态。

2. 组态打印作业

报表编辑器预先组态了用于输出表格形式变量记录报表的打印作业，作业名称为“Report Tag Logging RT Tables New”。该打印作业与布局“@ CCTlgRtTables. RPL”建立了链接，用户只需在此打印作业上进行打印机设置等组态即可。

在 WinCC 项目管理器的浏览窗口选择“报表编辑器”，将显示“布局”和“打印作业”条目。选择“打印作业”条目，则所有预先组态的打印作业都在数据窗口显示。双击名称为“Report Tag Logging RT Tables New”的打印作业，打开“打印作业”属性对话框，选择“打印机设置”选项卡，选择实际所需的打印机，单击“确定”按钮，完成组态，并退出对话框。

## 7.4　组态技巧与技术问答

本节将介绍一些 WinCC 组态报表的技巧，并针对组态过程中遇到的一些常见的技术问题进行分析和解答。

### 7.4.1　项目文档的输出方法

前文已述及，项目文档报表对应的是各 WinCC 编辑器的组态数据。项目文档报表的布局和打印作业均已预先组态好，用户可从各种 WinCC 编辑器中启动报表的输出。

在各种 WinCC 编辑器的文件菜单中，可找到表 7-5 所示的菜单命令。

表 7-5　与项目文档输出有关的菜单命令

| 菜单命令 | 功　　能 | 特　　征 |
|---|---|---|
| 打印项目文档 | 立即打印报表。将使用在打印作业中设置的布局。报表将输出到在打印作业属性对话框中所设置的打印机/文件 | 不显示打印机选择对话框，立即启动打印。根据项目大小的不同，该过程可能要花费一些时间。当正在从系统读取数据时，该过程不能取消 |
| 查看项目文档 | 打开要输出的报表的预览 | 根据项目大小的不同，该过程可能要花费一些时间。正在从系统读取数据时，该过程不能取消 |
| 项目文档设置… | 打开打印作业属性对话框。可指定所要使用的布局、报表将要输出到的打印机或文件，以及将要输出的页面 | |

输出项目文档的步骤如下：

1）在 WinCC 项目管理器中，通过在菜单栏或浏览窗口中双击弹出式菜单，可打开期望输出项目文档的 WinCC 编辑器，例如图形编辑器、报警记录编辑器等。

2）在已经打开的编辑器中，选择“文件”菜单中的“打印项目文档”菜单命令。

如果知道项目文档所使用的打印作业，可通过直接在 WinCC 项目管理器中启动相应的打印作业输出该项目文档。

1）选择 WinCC 项目管理器的浏览窗口中的“报表编辑器”条目，显示“布局”和“打印作业”条目。

2）单击浏览窗口中的“打印作业”条目，现有的所有打印作业均显示在浏览窗口中。

3）用鼠标右键单击相应的打印作业，在弹出的菜单中选择“打印”命令，将依照该打印作业链接的布局打印项目文档。

在 WinCC 项目管理器启动项目文档的打印时，不显示打印机选择对话框，而立即启动打印。根据项目大小的不同，该过程可能要花费一些时间。当正在从系统读取数据时，该过程不能取消。

在某些编辑器中，从应用程序中输出项目文档与通过 WinCC 项目管理器中的打印作业直接启动项目文档的输出不同。

## 7.4.2　输出硬拷贝（画面拷屏）的方法

WinCC 可以通过两种方法输出硬拷贝：

- 通过组合键输出硬拷贝。
- 通过报表对象输出硬拷贝。

### 1. 通过组合键输出硬拷贝

在 WinCC 项目管理器中，可使用项目属性来指定用于输出硬拷贝的组合键。这样可以创建整个画面、部分画面或当前窗口的硬拷贝。为此，必须将命令行参数传递给应用程序“PrintScreen”。与通过报表对象输出硬拷贝相反，通过应用程序“PrintScreen”的输出将直接传送到默认打印机，不使用已定义的页面布局。启动输出可有两种方式：

- 通过项目属性中所组态的组合键。
- 通过启动计算机启动列表中的“PrintScreen”应用程序。

在启动列表中，可传递输出的参数，启动输出的上述两种方式均应考虑参数。如果在项目属性和启动列表的参数中都组态了快捷键，则使用项目属性中的快捷键。然而，启动列表中的附加输出参数仍然应加以考虑。

（1）组态项目属性中的组合键

通过组合键的硬拷贝输出可在 WinCC 中全局设置，将直接输出到默认打印机。在这种情况下，无法从报表系统中获取布局，取而代之，可使用“PrintScreen”应用程序。组态组合键的步骤如下：

1）用鼠标右键单击 WinCC 项目管理器浏览窗口中的项目名称，在弹出的菜单中打开“项目属性”对话框。

2）选择“热键”选项卡，然后选择“动作”区中的“硬拷贝”，如图 7-18 所示。






图 7-18　组态输出硬拷贝的组合键

3）单击“分配”按钮下的文本框，并输入组合键，例如“Ctrl + P”，组合键将显示在文本框中。

4）通过单击“分配”按钮使设置生效，并关闭对话框。

在 WinCC 项目运行时，按下组合键〈Ctrl + P〉输出硬拷贝。

（2）在启动列表中添加“打印画面”应用程序

“打印画面”应用程序将被输入到计算机启动列表的“附加任务/应用程序”下，应用程序将带有命令行格式的参数，用于指定输出以及如何启动输出。组态步骤如下：

1）在 WinCC 项目管理器的浏览窗口中选择“计算机”。在数据窗口中，用鼠标右键单击要输出硬拷贝的计算机，然后在弹出的菜单中选择“属性”，打开“计算机属性”对话框。

2）选择“启动”选项卡，单击“添加...”按钮，打开“添加应用程序”对话框显示。

3）单击“浏览…”按钮，并浏览 WinCC 安装目录中的“bin”文件夹。选择此处的“PrtScr. exe”应用程序，并单击“打开”。

4）在“命令行参数”域中输入所期望的参数，如图 7-19 所示。




图 7-19　添加“PrinttScreen”应用程序到启动列表

5）“PrintScreen”应用程序不需要组态“工作目录”和“打开时的窗口”域。

6）使用“确定”按钮对输入进行确认。

图7-19中，命令行参数“-C”表示激活WinCC时启动PrtScr.exe，然后立即启动指定区域的硬拷贝，接着再终止PrtScr.exe。“-C=50；50；300；400”表示立即启动起始位置为上起50个像素、左起50个像素、结束位置为右起300个像素，下起400个像素区域的硬拷贝。“PrintScreen”应用程序命令行参数的设置可以参考WinCC帮助系统中“硬拷贝的输出参数”一节的相关内容。

**2. 通过报表对象输出硬拷贝**

为了在报表中输出硬拷贝，页面布局编辑器提供了“硬拷贝”报表对象，该报表对象允许创建整个画面、部分画面或当前窗口的硬拷贝。与通过组合键输出硬拷贝相反，这种方式可在已定义的页面布局中输出。也可以将日志对象插入到不同的报表中，并与其他数据一起输出。

通过报表对象输出硬拷贝的方法和步骤如下：

1）创建新的页面布局，并在页面布局编辑器中将其打开。

2）在对象选项板中选择“标准对象”→“动态对象”→“硬拷贝”插入到页面布局的工作区中，并将其拖动到需要的大小。

3）双击添加的“硬拷贝”对象，打开对象属性对话框，并选择“连接”选项卡。

4）选中“区域选择”后，单击“编辑”按钮或双击“区域选择”，打开“区域选择”对话框，允许选择将要输出的画面。

5）在对话框的“区域”框内的下拉列表框中有“整个屏幕的硬拷贝”、“当前窗口的硬拷贝”和“一个分段的硬拷贝”三个选项。如果选择“一个分段的硬拷贝”，则必须在对话框的“一个段的硬拷贝”框内的文本框中指定区域的左上角位置（X、Y坐标）和以像素为单位的区域大小（宽度和高度），如图7-20所示。

6）单击“确定”按钮关闭对话框，并保存布局。

7）创建打印作业，并选择与此处所组态的页面布局建立链接。

8）通过WinCC项目管理器中的打印作业或通过在WinCC画面中所组态的调用（例如按钮的动作）来启动报表的输出。




图7-20　“区域选择”对话框

在图7-20所示的“区域选择”对话框中，区域选择以及位置和大小的规范均可通过WinCC变量进行动态化。为此，可选择相关的“变量”前的复选框，并使用文件夹按钮来选择变量选择对话框中的变量。如果知道变量名称，还可直接输入。

### 7.4.3　组态在过程画面中启动报表输出的按钮的方法

如果希望在WinCC运行系统的过程画面中单击按钮来启动报表的打印，可以通过在



画面中组态一个按钮的 C 动作，在 C 动作中调用用于报表打印的标准函数来实现。详细组态方法和步骤如下：

1）在图形编辑器中打开期望的过程画面，在过程画面的相关位置添加一个对象选项板中的“标准”→“窗口对象”→“按钮”。

2）双击按钮或用鼠标右键单击按钮，然后在弹出的菜单中选择“属性”，打开该按钮的“对象属性”对话框。选择“事件”选项卡，在左侧的窗口中选择“鼠标”，在右侧窗口“按左键”的动态列的白色闪电上单击鼠标右键，在弹出的菜单中选择“C 动作...”，打开 C 动作编辑器，并在代码区中输入如下代码：

```
void OnLButtonDown(char * lpszPictureName, char * lpszObjectName, char * lpszPropertyName,
                   UINT nFlags, int x, int y)
{
    RPTJobPrint("PrintJob001");
}
```

3）编译 C 动作后，单击“确定”按钮，退出 C 动作编辑器，在图形编辑器中单击“保存”按钮，保存所作的组态。

在上述 C 代码中，标准函数 RPTJobPrint 用于输出报表，其参数“PrintJob001”是在 WinCC 报表编辑器中创建的打印作业的名称。还可以调用标准函数 RPTJobPreview 来实现打印预览。

### 7.4.4 将 WinCC 归档的历史数据导出到 Excel CSV 文件的方法

从 WinCC V6.2 开始，所有的历史数据通过算法进行了压缩处理，无法通过标准的 SQL 语句访问，需要在安装 WinCC 选件 WinCC/Connectivity Pack 的基础上，根据 WinCC/Connectivity Pack 的语法编写程序访问历史归档值和历史报警信息。或者通过 WinCC 欧洲版选件 WinCC Industrial Data Bridge 的组态，将 WinCC 的历史数据和实时数据写入到数据库或 Excel CSV 文件。选件 WinCC/Connectivity Pack 和 WinCC Industrial Data Bridge 都需要另外购买授权。

但是，对于生产数据的分析，例如仅需要某一时间段的 Excel CSV 数据文件，可以通过以下三种方法非常快捷、方便地将历史数据导出到 Excel CSV 文件：

- 使用 WinCC 标准控件的工具栏按钮。
- 使用 Excel 模板文件。
- 使用“SQL Server Import/Export”向导。

**1. 使用 WinCC 标准控件的工具栏按钮**

（1）WinCC 在线趋势控件（WinCC Online Trend Control）

从 WinCC V6.2 版本开始，WinCC 在线趋势控件的工具栏增加了“报表保存”按钮。在 WinCC 画面中添加 WinCC 在线趋势控件后，双击该控件打开“WinCC 在线趋势控件的属性”对话框，选择“工具栏”选项卡，在“键盘功能”列表框中选中“保存报表”前的复选框。激活运行系统后，WinCC 在线趋势控件的最后一个工具栏按钮显示为存盘图标，如图 7-21 所示。

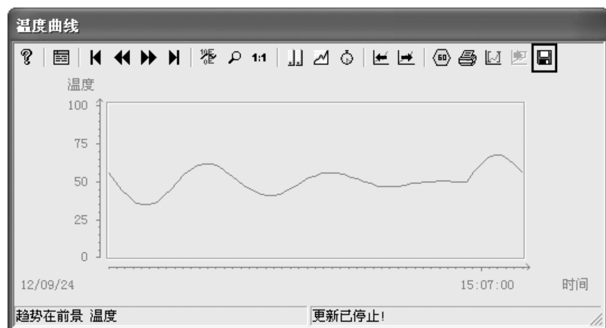
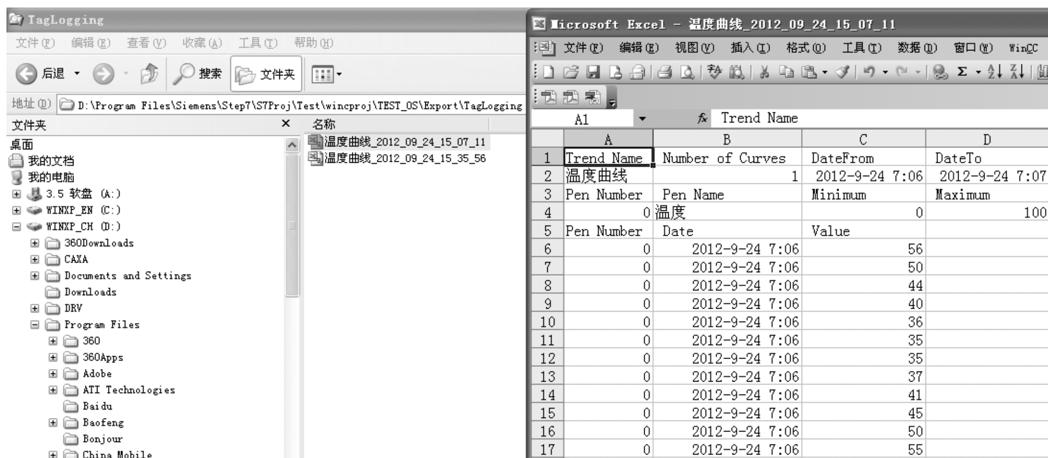



图 7-21　WinCC 在线趋势控件的“保存报表”按钮

将历史数据保存到 Excel CSV 文件中，必须先单击工具栏按钮停止趋势更新，此时“启动/停止更新”工具栏按钮显示为，同时“保存报表”按钮有效。单击“保存报表”按钮，变量记录系统的历史数据报表将以“.csv”文件的格式保存在 WinCC 项目路径的“Export/TagLogging”文件夹下，文件名由控件的窗口标题和报表生成的日期时间组成，如图 7-22 所示。




| | A | B | C | D |
|---|---|---|---|---|
| 1 | Trend Name | Number of Curves | DateFrom | DateTo |
| 2 | 温度曲线 | 1 | 2012-9-24 7:06 | 2012-9-24 7:07 |
| 3 | Pen Number | Pen Name | Minimum | Maximum |
| 4 | 0 | 温度 | 0 | 100 |
| 5 | Pen Number | Date | Value | |
| 6 | 0 | 2012-9-24 7:06 | 56 | |
| 7 | 0 | 2012-9-24 7:06 | 50 | |
| 8 | 0 | 2012-9-24 7:06 | 44 | |
| 9 | 0 | 2012-9-24 7:06 | 40 | |
| 10 | 0 | 2012-9-24 7:06 | 36 | |
| 11 | 0 | 2012-9-24 7:06 | 35 | |
| 12 | 0 | 2012-9-24 7:06 | 35 | |
| 13 | 0 | 2012-9-24 7:06 | 37 | |
| 14 | 0 | 2012-9-24 7:06 | 41 | |
| 15 | 0 | 2012-9-24 7:06 | 45 | |
| 16 | 0 | 2012-9-24 7:06 | 50 | |
| 17 | 0 | 2012-9-24 7:06 | 55 | |

图 7-22　单击“保存报表”按钮输出的 CSV 格式的历史数据

单击“保存报表”按钮后，报表会自动创建并保存，但是没有任何提示对话框。

需要注意的是，图 7-21 中 WinCC 在线趋势控件显示的数据的时间范围是 15：06 ~ 15：07，但是到处文件中数据的时间范围是 7：06 ~ 7：07，这时因为 WinCC 数据库保存的时间为 UTC 格林尼治标准时间，因此会比北京时间少 8 个小时。

WinCC 在线表格控件从 WinCC V7.0 版本才有将历史数据导出到 Excel CSV 文件的功能，是在项目激活时通过单击工具栏的“导出数据”按钮实现的，如图 7-23 所示。




图 7-23　WinCC 在线表格控件的“导出数据”按钮



停止 WinCC 在线表格控件的更新后，单击“导出数据”按钮，将会打开一个“导出数据”对话框，在对话框中可以查看导出设置并启动导出，还可以选择导出的文件和导出文件的存放路径，如图 7-24 所示。




图 7-24　“导出数据”对话框

2. 使用 Excel 模板

在西门子技术网站上提供了一个标准的 Excel 模板文件，可以方便地将 WinCC 的归档数据导出到 Excel CSV 文件。在浏览器（例如微软的 Internet Explorer）中输入西门子技术网站的网址 http：//support. automation. siemens. com/CN/view/zh/35840700，将打开名称为“使用 WinCC OLE DB Provider 导出 WinCC / CAS 归档数据”的技术文档的网页，在该网页的下载列表中找到名称为“OLE – DB Tag and Alarm Export. xls”的 Excel 应用程序文件，如图 7-25 所示。单击“下载”列的图标，按照提示将该 Excel 文件下载到计算机的合适位置，解压后在 Microsoft Office Excel 2003 中用鼠标左键双击打开该文件。

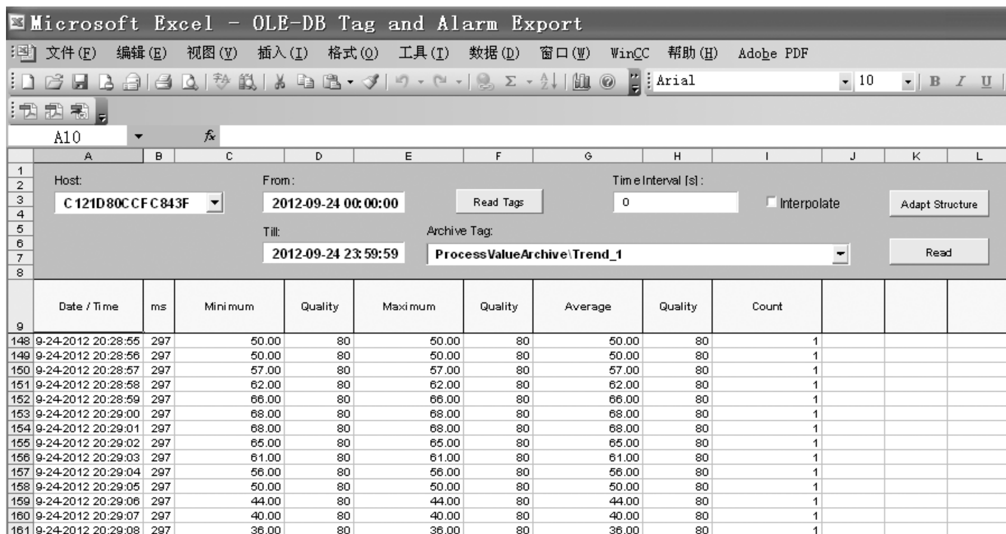



图 7-25　下载 Excel 模板的网页

在打开“OLE - DB Tag and Alarm Export. xls”的过程中，Microsoft Office Excel 2003 将弹出“安全警告”对话框，如图 7-26 所示，在对话框中单击“启用宏”按钮（如果 Excel 的安全级别为高，将无法启用宏，可以单击菜单“工具”→“宏”→“安全性”将安全级别设置为“中”），进入已经设计好的用于输出 WinCC 历史数据的 Excel 模板，如图 7-27 所示。




图 7-26　启用 Excel 模板中的宏




| | Date / Time | ms | Minimum | Quality | Maximum | Quality | Average | Quality | Count |
|---|---|---|---|---|---|---|---|---|---|
| 148 | 9-24-2012 20:28:55 | 297 | 50.00 | 80 | 50.00 | 80 | 50.00 | 80 | 1 |
| 149 | 9-24-2012 20:28:56 | 297 | 50.00 | 80 | 50.00 | 80 | 50.00 | 80 | 1 |
| 150 | 9-24-2012 20:28:57 | 297 | 57.00 | 80 | 57.00 | 80 | 57.00 | 80 | 1 |
| 151 | 9-24-2012 20:28:58 | 297 | 62.00 | 80 | 62.00 | 80 | 62.00 | 80 | 1 |
| 152 | 9-24-2012 20:28:59 | 297 | 66.00 | 80 | 66.00 | 80 | 66.00 | 80 | 1 |
| 153 | 9-24-2012 20:29:00 | 297 | 68.00 | 80 | 68.00 | 80 | 68.00 | 80 | 1 |
| 154 | 9-24-2012 20:29:01 | 297 | 68.00 | 80 | 68.00 | 80 | 68.00 | 80 | 1 |
| 155 | 9-24-2012 20:29:02 | 297 | 65.00 | 80 | 65.00 | 80 | 65.00 | 80 | 1 |
| 156 | 9-24-2012 20:29:03 | 297 | 61.00 | 80 | 61.00 | 80 | 61.00 | 80 | 1 |
| 157 | 9-24-2012 20:29:04 | 297 | 56.00 | 80 | 56.00 | 80 | 56.00 | 80 | 1 |
| 158 | 9-24-2012 20:29:05 | 297 | 50.00 | 80 | 50.00 | 80 | 50.00 | 80 | 1 |
| 159 | 9-24-2012 20:29:06 | 297 | 44.00 | 80 | 44.00 | 80 | 44.00 | 80 | 1 |
| 160 | 9-24-2012 20:29:07 | 297 | 40.00 | 80 | 40.00 | 80 | 40.00 | 80 | 1 |
| 161 | 9-24-2012 20:29:08 | 297 | 36.00 | 80 | 36.00 | 80 | 36.00 | 80 | 1 |

图 7-27　用于输出 WinCC 历史数据的 Excel 模板

利用图 7-27 所示的 Excel 模板导出 WinCC 历史数据的步骤如下：

1）在“Host”的下拉列表框中选择本地计算机的名称。

2）单击“Read Tags”按钮读取当前激活的 WinCC 项目中所有归档变量的名称。

3）通过“From”和“Till”下的文本框设置输出数据的时间范围，在“Archive Tag”的下拉列表中选择要输出的归档变量的名称。

4）单击“Read”按钮，需要导出的数据将会自动添加到 Excel 表格中，如图 7-27 所示。

图 7-27 中，每一条记录包含最小值、最大值、平均值和记录总数，归档变量 Trend _ 1 的归档周期为 1 秒，因此最小值、最大值、平均值都一样，记录总数为 1 条。可以根据需要在“Time Interval [S]”下的文本框中设定导出数据的时间间隔。



在 Excel 中对文件进行另存，即可以保存导出的数据。

3. 使用“SQL Server Import/Export”向导

利用“SQL Server Import/Export”向导组态，可以通过 WinCC OLE DB 提供的程序来访问 WinCC 数据库，并将数据导出到其他格式的文件中，例如 Excel CSV 文件、文本文件等。以下举例说明使用 SQL Server 2005 将 WinCC 的历史数据导出到 Excel CSV 文件的方法和详细步骤，例子中使用的 WinCC 版本为 V6.2 亚洲版。

1）打开 WinCC 变量记录系统，查询需要导出的变量的归档名称，如图 7-28 所示。




图 7-28 查询导出变量的归档名称

2）启动 WinCC，打开相应的 WinCC 项目并激活运行系统。

3）启动 SQL Server Management Studio。单击“开始”菜单，选择“所有程序”→“Microsoft SQL Server 2005”→“SQL Server Management Studio”，将弹出“Connect to Server”对话框，如图 7-29 所示。在“Server name”后的下拉列表框中选择“计算机名称\WINCC”，然后单击“Connect”按钮，将进入“Microsoft SQL Server Management Studio”界面。




图 7-29 “Connect to Server”对话框

4）在“Microsoft SQL Server Management Studio”界面中，选择 WinCC 项目对应的数据库，即 CC_项目名称_日期_时间 R，如图 7-30 所示。单击该数据库，在其树形结构中选择“Tables”→“db0. Archive”，用鼠标右键单击“db0. Archive”，在弹出的菜单中选

择“Open Table”，在右边的窗口中将显示表“db0. Archive”的数据。此时可以根据 ValueName（变量名称）查询变量对应的 ValueID，例如归档 ProcessValueArchive 中归档变量 Trend_ 1 的 ValueID 为 8。

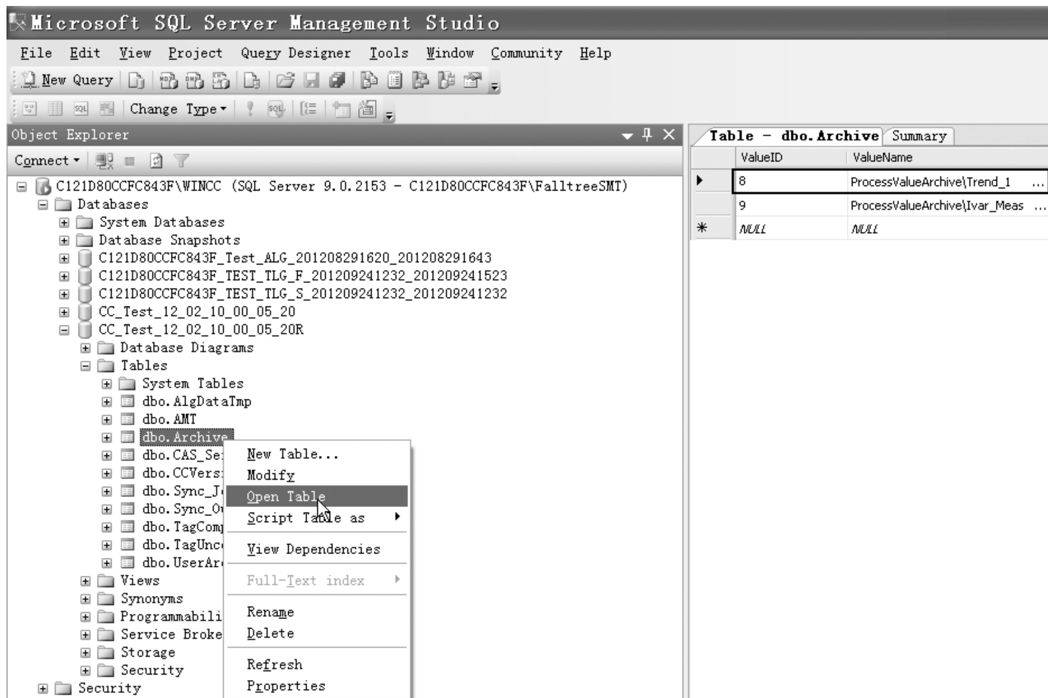



图 7-30　在“Microsoft SQL Server Management Studio”界面中查询数据库

5）在“Microsoft SQL Server Management Studio”界面中用鼠标右键单击 WinCC 运行数据库名称，例如图 7-30 中的“CC_Test_12_02_10_00_05_20R”，在弹出的菜单中选择“Tasks”→“Exoport Data…”，将打开“SQL Server Import and Export Wizard”对话框。单击“Next”按钮，将进入向导的“Choose a Data Source”（选择一个数据源）部分，如图 7-31所示。

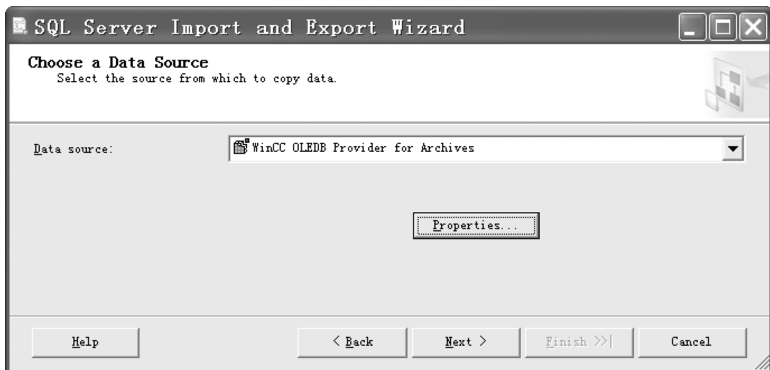



图 7-31　选择输入/输出数据向导的数据源



6）在“Data Source：”后的下拉列表框中，选择数据源“WinCC OLEDB Provider for Archives”。单击“Properties…”按钮，打开“数据链接属性”对话框，来设置数据源的属性，如图 7-32 所示。

7）在图 7-32 所示对话框的“数据源”后的文本框中输入“. \ WinCC”，在“输入要使用的初始目录”的下拉列表框中输入运行数据库名称，例如“CC_Test_12_02_10_00_05_20R”，单击“确定”按钮完成参数设置并回到向导对话框。单击“确定”按钮后，有时会出现初始化失败的错误提示，在此单击“Properties”按钮进入“数据链接属性”对话框，已经输入的参数丢失。需要激活 WinCC 运行系统，重复图 7-32 中的设置，直到输入的参数被保留，确认后没有错误提示。

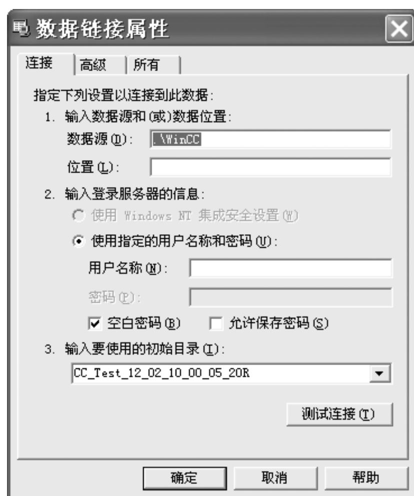



图 7-32　“数据链接属性”对话框

8）在向导对话框中单击“Next”按钮，将进入向导的“Choose a Destination”（选择一个输出目标）部分，设置数据导出的参数，如图 7-33 所示。在“Destination”后的下拉列表框中选择“Flat File Destination”。单击“File name”后的“Browse…”按钮，在打开的对话框中设置输出文件的格式和存放位置。选中“Column names in the first data row”前的复选框。




图 7-33　设置数据导出的参数

9）单击“Next”按钮，进入向导的“Specify Table Copy or Query”（确定复制整个表格的数据还是根据查询结果输出）部分，如图 7-34 所示，选择“Write a query to specify the data to transfer”（编写查询语句来确定传输的数据）。

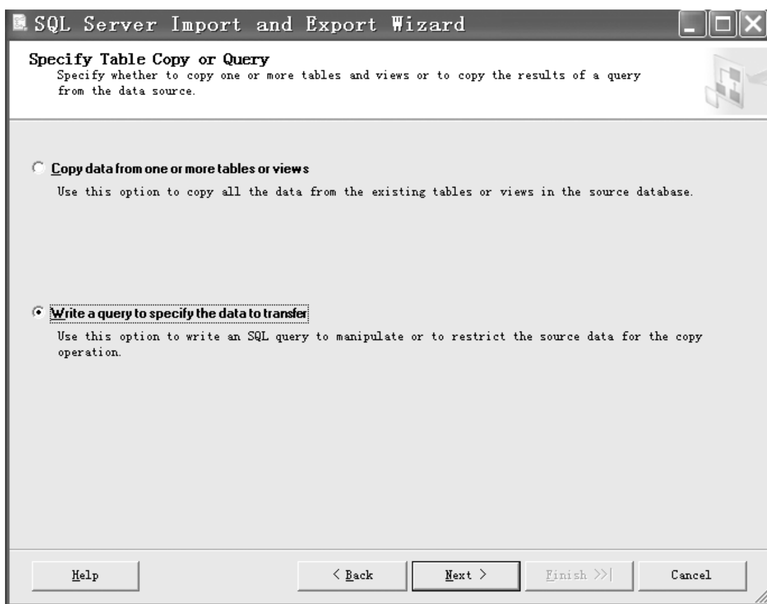



图7-34　设置导出数据的方式

10）单击“Next”按钮，进入向导的“Provide a Source Query”（提供查询语句）部分，如图7-35所示，在“SQL statement”下的文本框中输入SQL查询语句：TAG：R，8，‘0000－00－00 00：00：00.000’，‘0000－00－00 00：01：00.000’。




图7-35　输入SQL查询语句

图7-35中的SQL查询语句格式为WinCC/Connectivity Pack语法，并不是标准的SQL语句，语法为：TAG：R，<ValueID>，<TimeBegin>，<TimeEnd>。其中<ValueID>为变量的ValueID。<TimeBegin>为开始时间，<TimeEnd>为结束时间，格式均为



'YYYY – MM – DD hh：mm：ss. msc'。由于 WinCC 数据库保存的时间为 UTC 格林尼治标准时间，会比北京时间少 8 个小时。该 SQL 查询语句还支持相对时间间隔的选择：<TimeBegin> = '0000 – 00 – 00 00：00：00. 000'代表数据库里的第一条记录，<TimeEnd> = '0000 – 00 – 00 00：00：00. 000'代表数据库里的最后一条记录。图 7–35 的 SQL 语句表示从数据库的第一条记录开始一分钟以内的数据。

表 7–6 为 SQL Server2005 数据库中保存 WinCC 归档数据的表的结构。

**表 7–6　SQL Server2005 数据库中保存 WinCC 归档数据的表的结构**

| 域　名　称 | 类　　型 | 注　　释 |
|---|---|---|
| ValueID | 整型（4 B）或整型（8 B） | 值的唯一表示，长度取决于查询的类型 |
| TimeStamp | 日期时间 | 时间戳 |
| RealValue | 实型（8 B） | 变量值 |
| Quality | 整型（4 B） | 变量的质量代码 |
| Flags | 整型（4 B） | 内部控制参数 |

11）在图 7–35 所示的对话框中单击"Next"按钮，进入向导的"Configure Flat File Destination"（组态目标平面文件）部分，如图 7–36 所示。在"Column delimiter"（列分隔符）后的下拉列表框中选择"Semicolon {;}"（分号）作为目标文件中表格的列分隔符。单击"Preview…"按钮可以预览查询的结果。单击"Edit transform…"（编辑转换）可以打开"Column Mappings"（列映射）对话框，如图 7–37 所示，在这里可以指定 CSV 文件的数据格式，完成后单击"OK"按钮，退出该对话框。

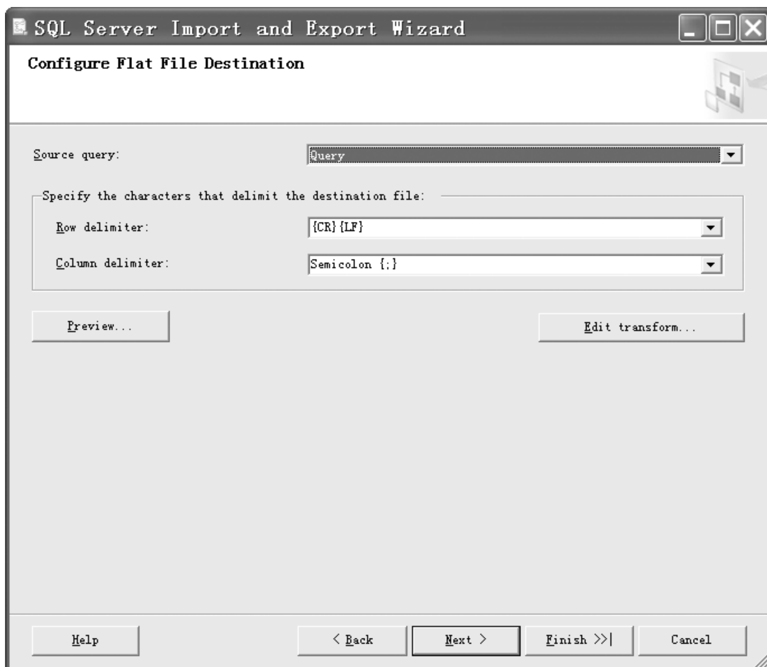



图 7–36　组态目标文件的属性

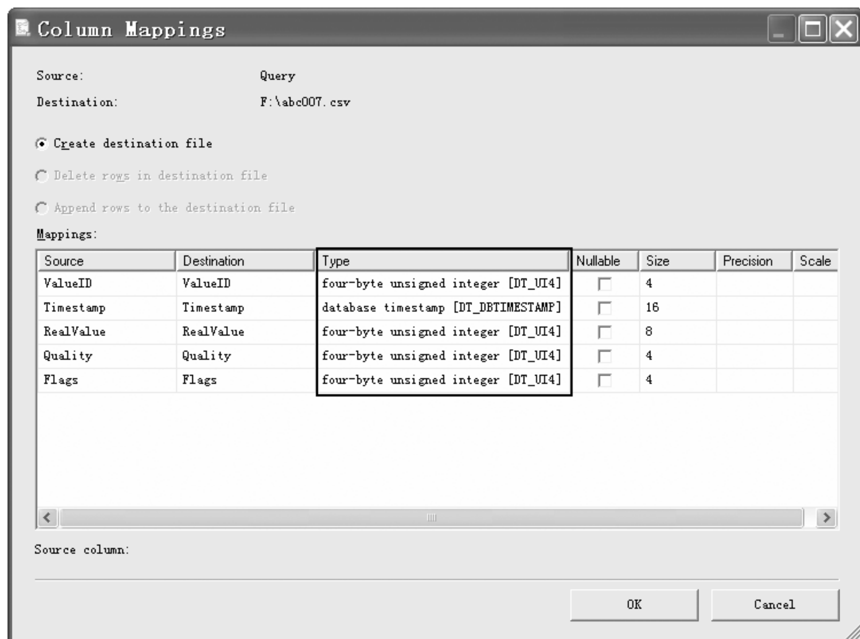



图 7-37　“列映射”对话框

12）在图 7-36 所示的对话框中单击“Next”按钮，在接下来的对话框中选择以“无保护”（Do not save sensitive data）级别，在 Windows 文件系统生成 SSIS（SQL Server Integration Services，SQL Server 集成服务）package，如图 7-38 所示。

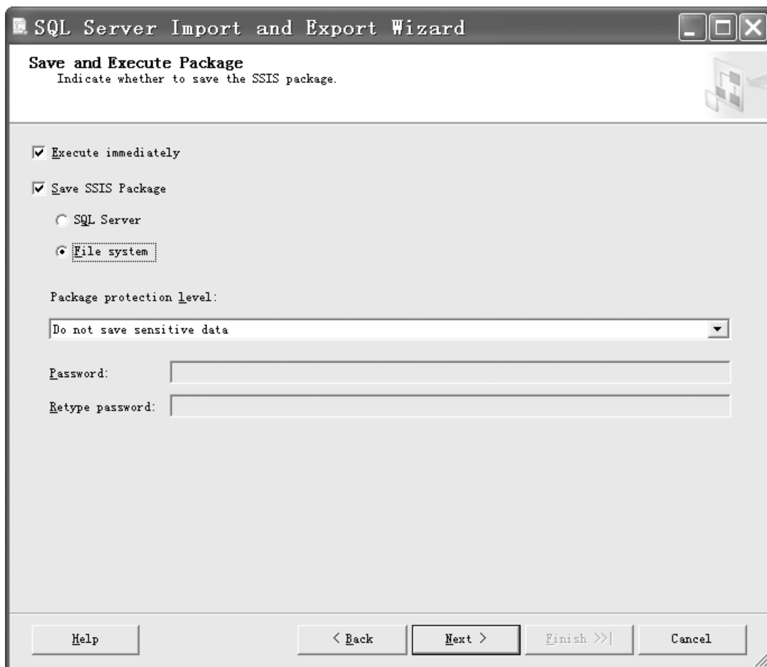



图 7-38　组态保存 SSIS package 的方式



13）单击“Next”按钮，在接下来的对话框中组态 SSIS package 的文件名和保存路径，如图 7-39 所示。




图 7-39　组态 SSIS package 的文件名和保存路径

14）单击“Next”按钮，在打开的对话框中单击“Finish”按钮，开始导出数据。图 7-40 显示导出的结果，如果显示在“Setting Destination Connection”处出错，则需要重新设定图 7-37 中的参数。

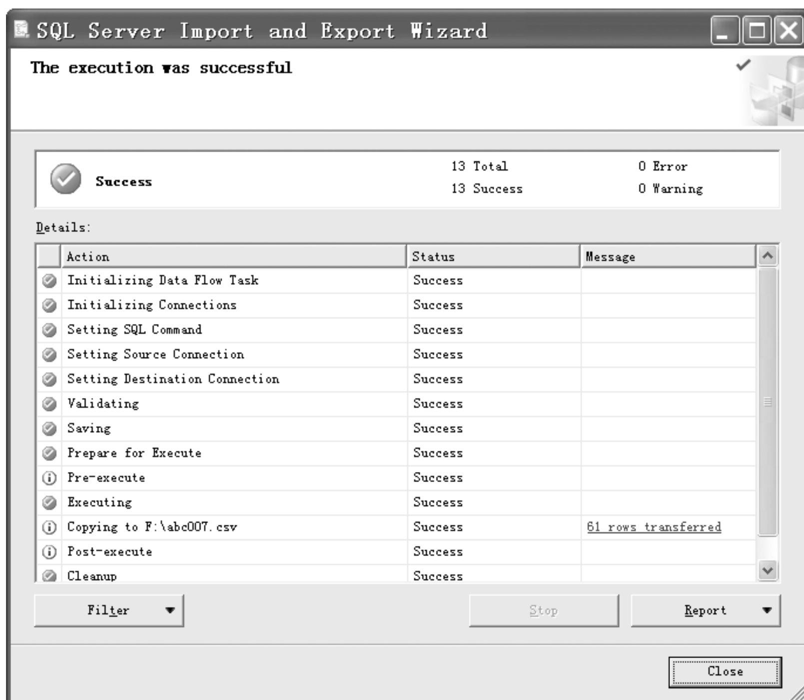



图 7-40　导出结果

图 7-41 为双击保存的 SSIS pacage "ID8. dtsx" 后的结果，单击 "Execute" 按钮可以重复执行数据导出的过程。首次执行数据导出后，生成文件 "F：\ abc007. csv"，将该文件剪切保存到其他文件夹下，在需要的时候再次执行 "ID8. dtsx"，可以随时导出数据到文件 "F：\ abc007. csv" 中，当然前提是 WinCC 项目处于运行状态。

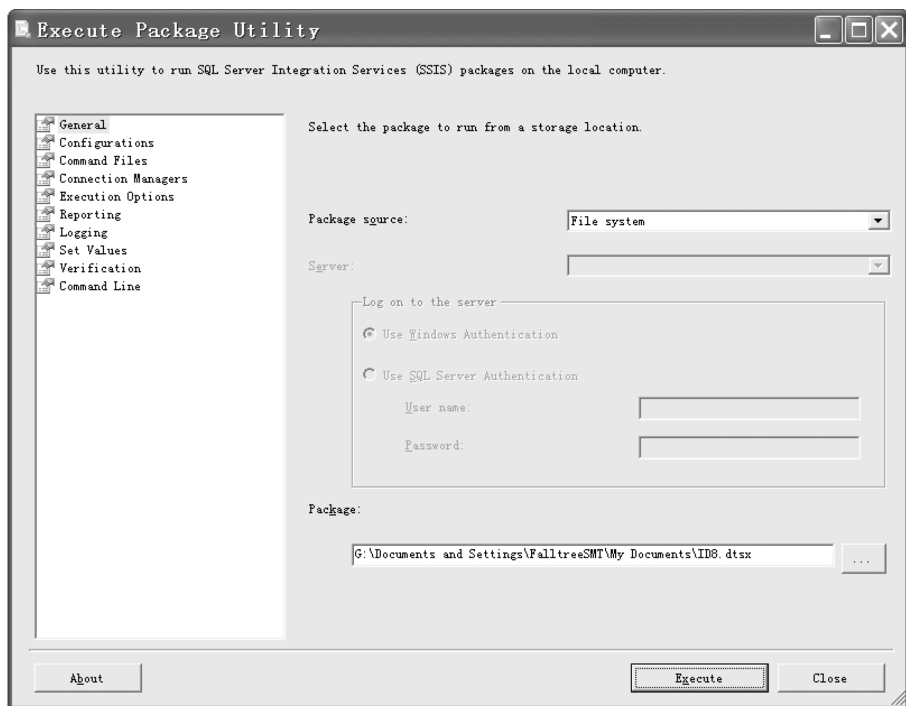



图 7-41　运行 SSIS pacage 的结果

如图 7-42 所示，打开导出的文件 "F：\ abc007. csv"，选中 A 列，选择菜单数据→ "分列"。在打开的 "文本分列向导 -3 步骤 1" 对话框中选择激活 "分隔符号" 后，单击 "下一步" 按钮，打开 "文本分列向导 -3 步骤 2" 对话框，如图 7-43 所示。在这里选择分隔符号为 "Tab 键" 和 "分号"，单击 "完成" 按钮，数据表将从 1 列分成 5 列，如图 7-44 所示。

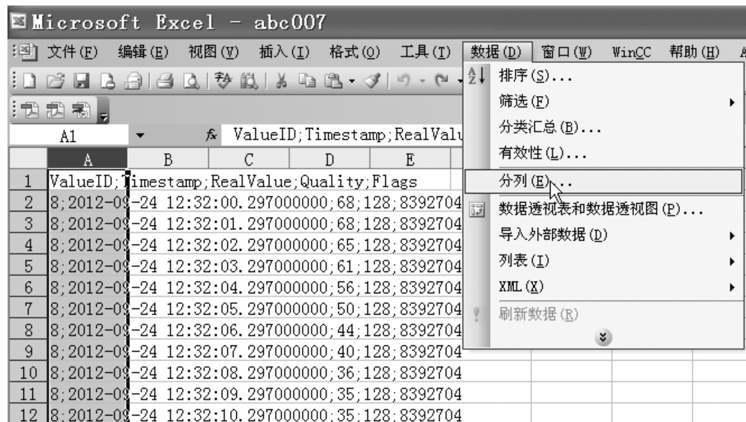



图 7-42　为导出的数据分列






图 7–43　选择分列的分隔符

Microsoft Excel - abc007

E1　Flags

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | ValueID | Timestamp | RealValue | Quality | Flags | | | |
| 2 | 8 | 32:00.3 | 68 | 128 | 8392704 | | | |
| 3 | 8 | 32:01.3 | 68 | 128 | 8392704 | | | |
| 4 | 8 | 32:02.3 | 65 | 128 | 8392704 | | | |
| 5 | 8 | 32:03.3 | 61 | 128 | 8392704 | | | |
| 6 | 8 | 32:04.3 | 56 | 128 | 8392704 | | | |
| 7 | 8 | 32:05.3 | 50 | 128 | 8392704 | | | |
| 8 | 8 | 32:06.3 | 44 | 128 | 8392704 | | | |
| 9 | 8 | 32:07.3 | 40 | 128 | 8392704 | | | |
| 10 | 8 | 32:08.3 | 36 | 128 | 8392704 | | | |
| 11 | 8 | 32:09.3 | 35 | 128 | 8392704 | | | |
| 12 | 8 | 32:10.3 | 35 | 128 | 8392704 | | | |

图 7–44　分列后的数据表

选中 B 列的所有数据（除列标题“Timestamp”），单击鼠标右键，在弹出的菜单中选择“设置单元格格式”命令，弹出“单元格格式”对话框，如图 7–45 所示。




图 7–45　设置“Timestamp”列的单元格格式

在“单元格格式”对话框中选择“数字”选项卡，在“分类”下的列表框中选择“自定义”，在“类型”下的文本框中输入“yyyy - mm - dd hh：mm：ss.000”，单击“确定”按钮退出对话框，“Timestamp”列的数据将按照设置的日期时间格式显示，如图 7-46 所示。

Microsoft Excel - abc007

文件(F) 编辑(E) 视图(V) 插入(I) 格式(O) 工具(T) 数据(D) 窗口(W) WinCC 帮助(H)

H19

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | ValueID | Timestamp | RealValue | Quality | Flags | | |
| 2 | 8 | 2012-09-24 12:32:00.297 | 68 | 128 | 8392704 | | |
| 3 | 8 | 2012-09-24 12:32:01.297 | 68 | 128 | 8392704 | | |
| 4 | 8 | 2012-09-24 12:32:02.297 | 65 | 128 | 8392704 | | |
| 5 | 8 | 2012-09-24 12:32:03.297 | 61 | 128 | 8392704 | | |
| 6 | 8 | 2012-09-24 12:32:04.297 | 56 | 128 | 8392704 | | |
| 7 | 8 | 2012-09-24 12:32:05.297 | 50 | 128 | 8392704 | | |
| 8 | 8 | 2012-09-24 12:32:06.297 | 44 | 128 | 8392704 | | |
| 9 | 8 | 2012-09-24 12:32:07.297 | 40 | 128 | 8392704 | | |
| 10 | 8 | 2012-09-24 12:32:08.297 | 36 | 128 | 8392704 | | |
| 11 | 8 | 2012-09-24 12:32:09.297 | 35 | 128 | 8392704 | | |
| 12 | 8 | 2012-09-24 12:32:10.297 | 35 | 128 | 8392704 | | |

图 7-46　设置单元格格式后的“Timestamp”列

### 7.4.5　WinCC 组态复杂报表的方法

在工业生产中数据报表一直占有非常重要的部分，它一般用来记录现场的工艺参数和统计信息。早期是由人工抄录然后统计出相关的报表。进入自动化控制阶段，报表可以由组态软件来实现。对于工业自动化控制领域的工程师来说，如何开发适合用户需求的报表一直是个难题。

本节将从报表的需求分析以及 WinCC 报表系统实现复杂报表的组态方法等方面进行详细介绍。最后总结出 WinCC 报表系统在应用中的优缺点和常见的问题的解决方法。

**1. 报表的需求分析**

从需求上来讲报表一般分为：内容上的需求、样式上和打印任务上的需求。

（1）内容需求

报表的内容上一般涉及数据的求和、平均、最大值、最小值等计算，同时也会涉及关系的处理，比如班组相关对应信息的统计分析等。表 7-7 是某厂锅炉专业班组技术经济小指标月报表，从内容上讲是一个典型的工业现场报表应用。

**表 7-7　锅炉专业班组技术经济小指标月报表**

| 名　　称 | 主蒸汽温度 TE - 5113 /℃ | 主蒸汽温度合格率 /% | 主蒸汽压力 PT - 5105 /MPa | 主蒸汽压力合格率 /% | 制粉系统运行时间 /h | 左烟汽氧量 AET - 5201A /% | 右烟汽氧量 AET - 5201B /% | 左低预出口烟温 TE - 5209A /℃ | 右低预出口烟温 TE - 5209B /℃ |
|---|---|---|---|---|---|---|---|---|---|
| 一班平均 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 一班最高值 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 一班最低值 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |



（续）

| 名　　称 | 主蒸汽温度 TE－5113 /℃ | 主蒸汽温度合格率 /% | 主蒸汽压力 PT－5105 /MPa | 主蒸汽压力合格率 /% | 制粉系统运行时间 /h | 左烟汽氧量 AET－5201A /% | 右烟汽氧量 AET－5201B /% | 左低预出口烟温 TE－5209A /℃ | 右低预出口烟温 TE－5209B /℃ |
|---|---|---|---|---|---|---|---|---|---|
| 二班平均 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 二班最高值 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 二班最低值 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 三班平均 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 三班最高值 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 三班最低值 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 四班平均 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 四班最高值 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 四班最低值 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

表 7-7 在数据采集没有难度，唯一的难度在于四班三倒后，每个月班的出勤日是不一样的，这对于月报表来说，计算本月每个班的出勤日需要应用一些算法。此类报表在连续生产的行业需求很广泛，例如，钢铁冶金行业需要一天 24 小时不停运转的系统。

还有一种报表内容上是分总的关系，就是一部分是简单的数据记录，另一部分是对上一部分数据的统计，例如汇总、求和、平均等，如图 7-47 所示。

营材配进系统
新疆中石油管道有限公司
**配混线称重报表**
ByOaily.RPL

| 日期 | 批次数 | 总重量 | 主料 | 1#辅料 | 2#辅料 | 3#辅料 | 4#辅料 | 5#辅料 | 6#辅料 |
|---|---|---|---|---|---|---|---|---|---|
| 2008-03-02 | 79 | 9492.7500 | 7677.0500 | .0000 | 155.6500 | .0000 | .0000 | 115.6600 | 871.8300 |
| 2008-03-03 | 64 | 7893.2000 | 6392.9500 | .0000 | 129.2900 | .0000 | .0000 | 90.6700 | 615.2600 |
| 2008-03-04 | 9 | 2194.8000 | 1785.2500 | .0000 | 23.6800 | .0000 | .0000 | 35.2100 | 179.9000 |

营材配进系统
新疆中石油管道有限公司
**配混线称重报表**
ByOaily.RPL

**汇总统计：**

| 批次数： | 总重量 | 主料 | 1#辅料 | 2#辅料 | 3#辅料 | 4#辅料 | 5#辅料 | 6#辅料 |
|---|---|---|---|---|---|---|---|---|
| 152 | 19580.7600 | 15865.2500 | .0000 | 308.6200 | .0000 | .0000 | 241.5400 | 1666.9900 |

图 7-47　具有分总关系的报表

报表内容的另一种需求就是涉及混合排列的问题，一张表上既有分项数据，中间也涉及统计信息，同时也会添加一些实时的信息在上面，如图 7-48 所示的报表，中间涉及分项的记录，数据求平均，以及一些操作信息的记录，既有历史数据，也有实时信息。

当然更复杂的报表需求也是有的，比如说涉及一张报表上既有实时信息又有历史信息，同时还涉及趋势视图、报警信息等，用户自己可以灵活地定义报表的内容和样式。

报表在内容上的需求就本质来讲是对客户关心信息的总结和整理。内容的需求也就决定了报表的样式和可能采用的数据处理方式。

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | xx车间xx工序生产运行日报表 | | | | | | | | | | | |
| 2 | 20___年___月___日　星期___ | | | | | | | | | | | |
| 3 | 时间 | ___蒸氨塔 | | | | | | | | ( )脱氨水泵 | | 操作备注 |
| 4 | | 蒸氨塔 | | | | | | | 分缩后 | 废水冷却 | 出口压力 | |
| 5 | | 入塔气压 | 氨水流量 | 塔底压力 | 塔顶压力 | 塔底温度 | 塔顶温度 | 底槽液位 | 氨气温度 | 温度 | | |
| 6 | | (MPa) | (m3/h) | (KPa) | (KPa) | (℃) | (℃) | (n) | (℃) | (℃) | (MPa) | |
| 7 | 0 | | | | | | | | | | | |
| 8 | 1 | | | | | | | | | | | |
| 9 | 2 | | | | | | | | | | | |
| 10 | 3 | | | | | | | | | | | |
| 11 | 4 | | | | | | | | | | | |
| 12 | 5 | | | | | | | | | | | |
| 13 | 6 | | | | | | | | | | | |
| 14 | 7 | | | | | | | | | | | |
| 15 | 平均 | | | | | | | | | | | ( )班： |
| 16 | 8 | | | | | | | | | | | |
| 17 | 9 | | | | | | | | | | | |
| 18 | 10 | | | | | | | | | | | |
| 19 | 11 | | | | | | | | | | | |
| 20 | 12 | | | | | | | | | | | |
| 21 | 13 | | | | | | | | | | | |
| 22 | 14 | | | | | | | | | | | |
| 23 | 15 | | | | | | | | | | | |
| 24 | 平均 | | | | | | | | | | | ( )班： |
| 25 | 16 | | | | | | | | | | | |
| 26 | 17 | | | | | | | | | | | |
| 27 | 18 | | | | | | | | | | | |
| 28 | 19 | | | | | | | | | | | |
| 29 | 20 | | | | | | | | | | | |
| 30 | 21 | | | | | | | | | | | |
| 31 | 22 | | | | | | | | | | | |
| 32 | 23 | | | | | | | | | | | |
| 33 | 平均 | | | | | | | | | | | ( )班： |
| 34 | 备注 | 脱氨水含氨氮(g/ml):________碱液浓度：__________ | | | | | | | | | | |

图 7-48　复杂报表实例图

（2）报表样式和打印任务的需求

从报表的内容需求分析不难看出，报表在样式上的需求一种是格式固定、内容固定，即事先设计好报表的样式，然后在其上填写数据，这也是传统报表延续下来的一种方式。另一种就是格式和样式固定，但是数据信息不固定，可能返回很多的数据集，也可能是没有数据。这对报表系统的数据归纳能力提出了要求，但是对样式上没有特别明确的需求。最后一种比较复杂，即报表内容不固定、样式不固定，完全由用户自己设计和决定显示的内容。

报表的打印任务需求问题很简单，就是事件触发打印或者定时打印。更深层次的分析，报表其实就是数据的显示问题。早期因为没有存储设备，只能人手工抄写进行记录。现在有了计算机进行存储，则最终需要的就是数据的统计分析和显示。当然更高的需求是能将这些报表以电子文档的形式在网络上共享。很多企业可以接受不需要打印输出的报表，而最终关心的是报表中的数据可以如实地存储并可以方便地访问。

**2. WinCC 实现复杂报表的组态方法**

针对用户的报表需求，如何使用 WinCC 的报表系统实现复杂的报表呢？这里将详细介绍利用 WinCC 选件 Connectivity Pack 结合 WinCC 过程值归档实现复杂报表的方法。

（1）报表需求

现场实行四班三运转，每个月的月底对各个班次的数据情况进行统计分析，包括各个班次的最大值、最小值和平均值等。

（2）组态分析

难点：各个班次的上班时间不固定，每月的天数不固定。涉及最大值、最小值和平



均值的计算。

关键点：如何确立某班次和某一时间段的对应关系。

（3）WinCC 实现方法

首先要解决数据的存储（归档）问题。参考第 6 章“WinCC 变量记录系统”的相关内容可知，WinCC 的过程值归档中，对应一个过程变量可以生成多个归档变量，归档的启动和停止可以由事件进行控制。那么可以通过在 WinCC 用户管理器创建多个用户，当登录 WinCC 运行系统的用户发生变化时，过程值归档的归档变量相应发生变化，从而确定班次和数据的对应关系。如图 7-49 所示，在 WinCC 变量记录中组态了两个班次的归档变量，其对应的过程变量均为 ArcTag。其中，归档变量 ATagArc、Aavg、Amax 和 Amin 分别对应一班归档的过程变量 ArcTag 的实际值、平均值、最大值和最小值，BTagArc、Bavg、Bmax 和 Bmin 分别对应二班归档的过程变量 ArcTag 的实际值、平均值、最大值和最小值。




图 7-49　归档变量定义

两个班次的归档启动和停止设置如图 7-50 所示，其中二进制变量 AarcB 和 AarcE 分别用于启动和停止一班的数据归档，BarcB 和 BarcE 用于启动和停止二班的数据归档。




图 7-50　归档的启动和停止设置

在 WinCC 全局脚本编辑器（C 脚本）中创建如下全局动作（针对两个班次的代码，四个班次的类似）：

```
int gscAction( void )
{
    #pragma option(mbcs)
   if(! strcmp(GetTagChar("@CurrentUser"),"yiban"))
   {
      SetTagBit("AarcE",0); //Return - Type: BOOL
      SetTagBit("AarcB",1); //Return - Type: BOOL
      SetTagBit("BarcE",1); //Return - Type: BOOL
      SetTagBit("BarcB",0); //Return - Type: BOOL
   }
   else if(! strcmp(GetTagChar("@CurrentUser"),"erban"))
   {
      SetTagBit("BarcE",0); //Return - Type: BOOL
      SetTagBit("BarcB",1); //Return - Type: BOOL
      SetTagBit("AarcE",1); //Return - Type: BOOL
      SetTagBit("AarcB",0); //Return - Type: BOOL
    }
}
```

其中“yiban”、“erban”是在用户管理器中创建的用户，该全局动作通过 WinCC 内部变量@CurrentUser（相关内容参考本书第4章4.6节）的变化来触发，如图7-51所示。




图7-51　设置全局动作的触发器

这样在项目运行时一班的用户登录就触发 ATagArc 变量的归档，二班的用户登录就触发 BTagArc，从而达到分班次统计数据的目的。

需要注意的是，组态归档变量的属性时，在“过程变量属性”对话框的“参数”选项卡中，应分别为归档变量 ATagArc/BTagArc、Aavg/Bavg、Amax/Bmax 和 Amin/Bmin 选



择“实际值”、“平均值”、“最大值” 和 “最小值” 的处理方法。

其次要解决的问题是数据分析。在归档设计中，已经确定了数据和班次的对应关系，那么接下来要解决的问题就是数据的整理了。在这里需要借助于 WinCC 的选件 WinCC/Connectivity Pack 和 ADO/WinCC OLE－DB 接口来实现，通过它们可以实现访问 WinCC 的归档数据，具体实现方法请参考本书第 8 章 8.2.6 小节的相关内容。

最后就是布局设计。这个布局的设计原理很简单，就是把“变量” 添加到布局中。排列对齐就可以了，如图 7-52 所示（图中所有班次都是用的一班的数据，用户可以将其他班次的数据做相应的替代）。

Copyright © 1994-2007 by SIEMENS AG

SysProjectName

| 名称 | 主蒸汽温度 TE-5113 /℃ | 主蒸汽温度 TE-5113 /℃ | 主蒸汽温度 TE-5113 /℃ | 主蒸汽温度 TE-5113 /℃ | 主蒸汽温度 TE-5113 /℃ | 主蒸汽温度 TE-5113 /℃ | 主蒸汽温度 TE-5113 /℃ | 主蒸汽温度 TE-5113 /℃ |
|---|---|---|---|---|---|---|---|---|
| 一班平均 | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag |
| 一班最高值 | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag |
| 一班最低值 | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag |
| 二班平均 | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag |
| 二班最高值 | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag |
| 二班最低值 | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag |
| 三班平均 | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag |
| 三班最高值 | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag | AmaxTag |
| 三班最低值 | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag | AMinTag |
| 四班平均 | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag | AavgTag |

图 7-52　复杂数据报表的报表布局设计

# 第 8 章

# 全局脚本及开放性接口

所有的 SCADA/HMI 系统都提供相应的脚本语言用于组态对象的动作和功能。WinCC 提供了 ANSI－C 和 VBScript 两种脚本，它们都可以集成在图形运行系统和全局脚本运行系统。本书第 4 章讲述过程画面动态化时已经介绍了图形运行系统中 ANSI－C 和 VBScript 两种脚本的应用，本章将主要介绍全局脚本的应用。

WinCC 以 Microsoft 技术作为后盾，几乎集成了 Microsoft 所有的开放性技术，包括 ActiveX、DDE、OPC、VBA、VBS、OLE、API 以及 Microsoft 强大而高效的数据库 Microsoft SQL Server 等。通过这些开放性的接口，可以把一些应用程序集成到 WinCC 系统中，使得 WinCC 扩展应用更加方便简单。本章将简要介绍一些开放性接口在 WinCC 系统中的应用。

## 8.1　全局脚本

全局脚本是使用 ANSI－C 和 VBS 的过程和动作时的通用术语，用来组态一些对象的动作（触发函数）。在运行系统中，后台任务（例如打印日常报表、监控变量或完成指定画面的计算等）均将作为动作来完成，这些动作均由触发器来启动。

### 8.1.1　ANSI-C 脚本

WinCC 支持使用函数和动作使 WinCC 项目中的过程动态化。这些函数和动作以 ANSI－C 编写。在使用 ANSI－C 脚本前，先了解下面几个概念。

**1. 函数和动作**

动作由触发器激活，也就是由初始事件启动。函数没有触发器，作为动作的组件使用，并在动态对话框、变量记录和报警记录中使用，如图 8-1 所示。触发器的相关概念请参考本书第 4 章 4.2 节。

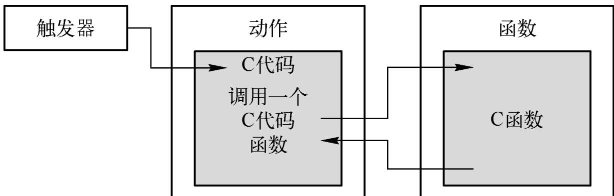



图 8-1　ANSI-C 的动作和函数



图 8-2 描述了函数和动作的使用范围。动作用于独立于画面的后台任务，例如打印日常报表、监控变量或执行计算等。函数是一段代码，可在多处使用，但只能在一个地方定义，WinCC 包括许多函数。此外，用户还可以编写自己的函数和动作。用户可以修改标准函数，重新安装或升级 WinCC 时，会删除修改过的标准函数，或使用未编辑的标准函数替换它们。因此，应在升级或重新安装之前事先备份修改过的函数。

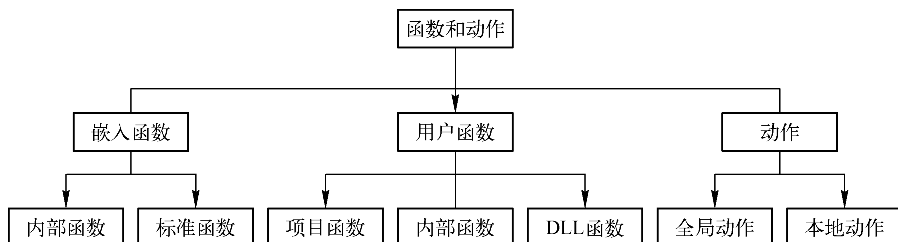



图 8-2　函数和动作的使用范围

2. 全局脚本编辑器

WinCC 通过全局脚本编辑器可支持函数和动作的创建与编辑，全局脚本编辑器从 WinCC 项目管理器的浏览窗口中启动。C-Script 全局脚本编辑器如图 8-3 所示。

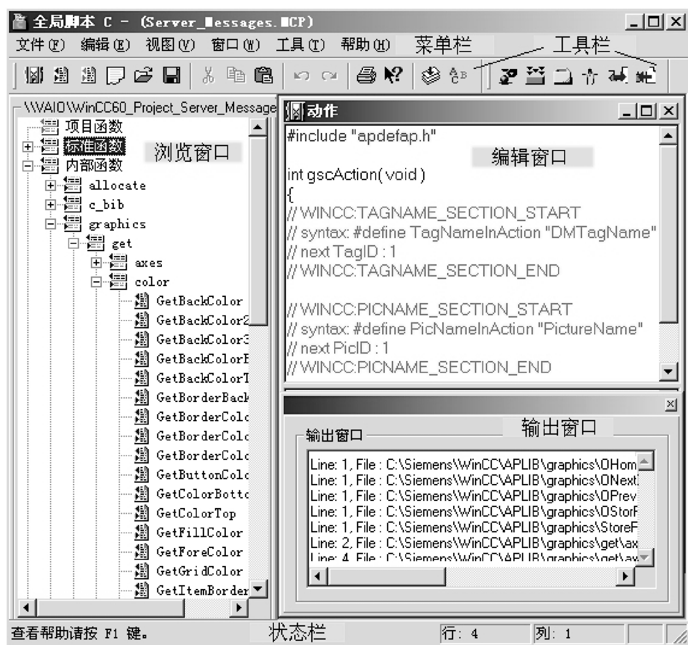



图 8-3　C-Script 全局脚本编辑器

C-Script 全局脚本编辑器的设计符合 Windows 标准，它具有工具栏、菜单栏和状态栏以及带有下拉菜单的多个窗口。

（1）浏览窗口

浏览窗口用于选择将要编辑或插入到编辑窗口中光标位置处的函数和动作。函数和动作均按组的多层体系进行组织。函数以其函数名显示，动作则显示文件名。

（2）编辑窗口

函数和动作均在编辑窗口中进行写入和编辑。只有在所要编辑的函数或动作已经打开时才显示编辑窗口。每个函数或动作都在单独的编辑窗口中打开，可同时打开多个编辑窗口。

（3）输出窗口

“在文件中查找”或“编译所有函数”的结果将显示在输出窗口中。默认状态下，输出窗口是可见的，但也可将其隐藏。

在文件中查找：搜索的结果按每找到一个搜索术语显示一行的方式显示在输出窗口中。每行均有一个行号，并会显示路径和文件名以及找到的搜索术语所在行的行号和文本。通过双击显示在输出窗口中的行，可打开相关的文件。光标将放置在找到搜索术语的行中。

编译所有函数：必要时，编译器将输出每个编译函数的警告和出错消息。下一行将显示已编译函数的路径和文件名以及编译器的摘要消息。

（4）菜单栏

菜单栏按钮根据情况而有所不同。它始终可见。

（5）工具栏

全局脚本编辑器有两个工具栏：标准工具栏和编辑工具栏。需要时可使其可见，并可使用鼠标拖动到屏幕的任何位置。表 8-1 列出了编辑工具栏的按钮的功能。

**表 8-1　全局脚本编辑器编辑工具栏按钮的功能表**

| 按　钮 | 功　能 | 组 合 键 |
|---|---|---|
| | 允许添加关于函数的信息，对于动作，则可以设置一个触发器。只有在编辑窗口打开时，此功能才可用 | 〈CTRL + I〉 |
| | 在活动编辑窗口中编译代码。只有在编辑窗口打开时，此功能才可用 | 〈SHIFT + F8〉 |
| | 打开变量对话框。只有在编辑窗口打开时，此功能才可用 | 〈CTRL + R〉 |
| | 打开用于选择画面的对话框。所选画面的名称将插入到编辑窗口中的光标位置处。只有在编辑窗口打开时，此功能才可用 | 〈CTRL + W〉 |
| | 导入动作。只有在活动窗口中存在动作时，此功能才可用 | 〈CTRL + M〉 |
| | 从活动编辑窗口中导出动作。只有在活动窗口中存在动作时，此功能才可用 | 〈CTRL + T〉 |

（6）状态栏

状态栏位于全局脚本窗口的下边缘，可以显示或隐藏。它包含了与编辑窗口中光标位置以及键盘设置等有关的信息。此外，状态栏可显示当前所选全局脚本函数的简短描述，也可显示其提示信息。

**3. 创建和编辑函数**

系统会区分项目函数、标准函数和内部函数，WinCC 带有可供广泛选择的标准函数和内部函数。此外，用户可以创建自己的项目函数和标准函数或修改标准函数。然而，需要注意的是，重新安装 WinCC 时，WinCC 包括的标准函数将被重写，所以任何函数修改都会丢失。不能创建或编辑内部函数。

如果在多个动作中必须执行同样的计算，只是具有不同的起始值，那么最好编写函



数来执行该计算。然后，可以在动作中用当前参数方便调用该函数，如图 8–4 所示。

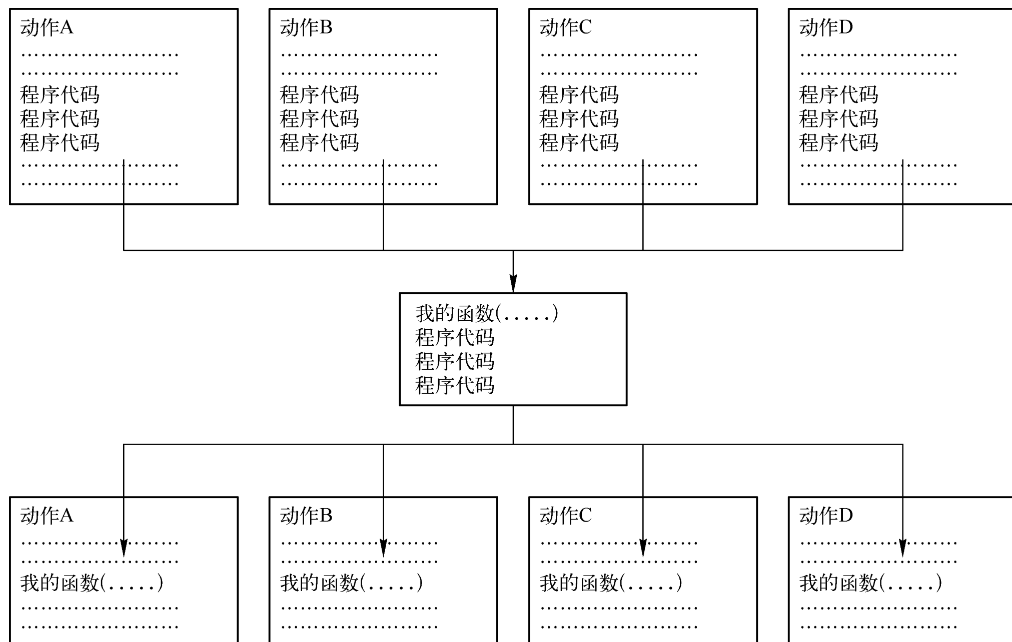



图 8–4　动作与函数使用结构图

这里以创建一个计算 3 个数的平均值的项目函数为例，详细介绍在 C–Script 全局脚本编辑器中创建和编辑函数的方法和步骤。

1）打开 C-Script 全局脚本编辑器。在 WinCC 项目管理器的浏览窗口单击“全局脚本”，将显示“C-Editor”和“VBS-Editor”，双击“C-Editor”或用鼠标右键单击“C-Editor”，然后在弹出的菜单中选择“打开”命令，将打开 C – Script 全局脚本编辑器。

2）新建项目函数。在 C – Script 全局脚本编辑器的浏览窗口用鼠标右键单击“项目函数”，在弹出的菜单中选择“新建”命令，将打开“编辑窗口”，同时在编辑窗口中新建一个名称为“New_Function”的项目函数，如图 8–5 所示。

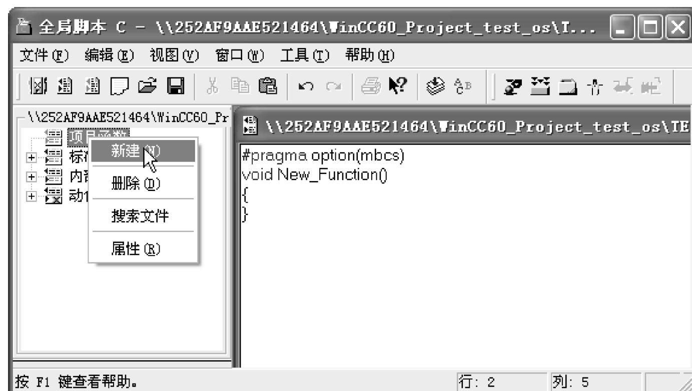



图 8–5　新建项目函数

3）将新建的项目函数的名称修改为“MeanValue”，并编写如下函数代码：

```
double MeanValue( double dValue1,double dValue2 ,double dValue3)
{
    double dMeanValue;
    dMeanValue = (  dValue1 + dValue2 + dValue3);
    return dMeanValue;
}
```

4）插入与函数有关的附加信息。单击编辑工具栏的按钮，打开“属性”对话框，如图 8-6 所示。选择需要的条目进行修改后，单击“确定”按钮确认修改，例如，勾选“口令”前的复选框，弹出“口令输入”对话框，在“口令”和“确认”框中输入期望的口令，可以为新建的函数设置访问保护。




图 8-6　函数“属性”对话框

5）编译保存函数。单击编辑工具栏中的按钮，对新建的函数进行编译，检查“输出”窗口输出的来自编译器的消息，如果编译器报告出错，则必须更正函数代码，完成后再次进行编译查错，直到编译正确为止。如图 8-7 所示。

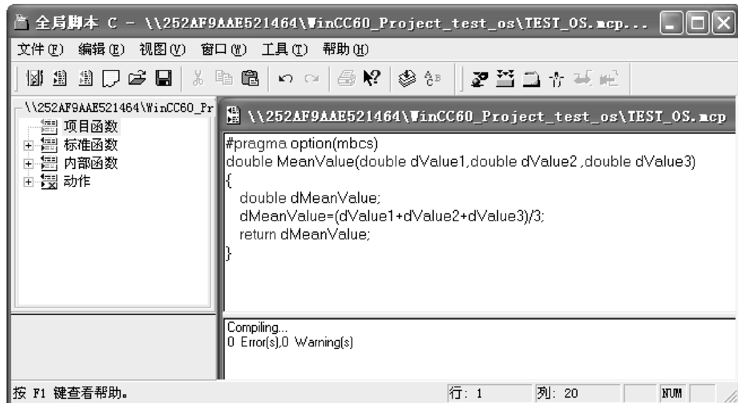



图 8-7　编译全局脚本



6）单击标准工具栏的▣按钮，保存新建的项目函数。

4. 创建和编辑动作

动作和函数之间的区别如下：

- 与函数相比，动作可以具有触发器。也就是说，函数在运行时不能由自己来执行。
- 动作可以导出和导入。
- 可为动作分配授权，该授权指的是全局脚本运行系统故障检测窗口的可操作选项。
- 动作没有参数。

动作分为图形运行系统的动作和全局动作，创建和编辑图形运行系统的动作的方法请参考本书第 4 章 4. 4. 6 节的相关内容。这里只介绍全局动作的创建和编辑。

以每隔 1 秒为名称为 Tag1 的 WinCC 变量（数据类型为无符号 8 位数）自加 1 为例详细介绍全局动作的创建方法和过程：

1）打开 C-Script 全局脚本编辑器。在 WinCC 的浏览窗口单击“全局脚本”，将显示“C-Editor”和“VBS-Editor”，双击“C-Editor”或用鼠标右键单击“C-Editor”，然后在弹出的菜单中选择“打开”命令，将打开 C-Script 全局脚本编辑器。

2）在 C-Script 全局脚本编辑器的浏览窗口用鼠标右键单击“动作”，在弹出的菜单中选择“新建”，或单击菜单“文件”→“新建动作”，都可以新建一个全局动作，同时打开“编辑窗口”，如图 8-8 所示。

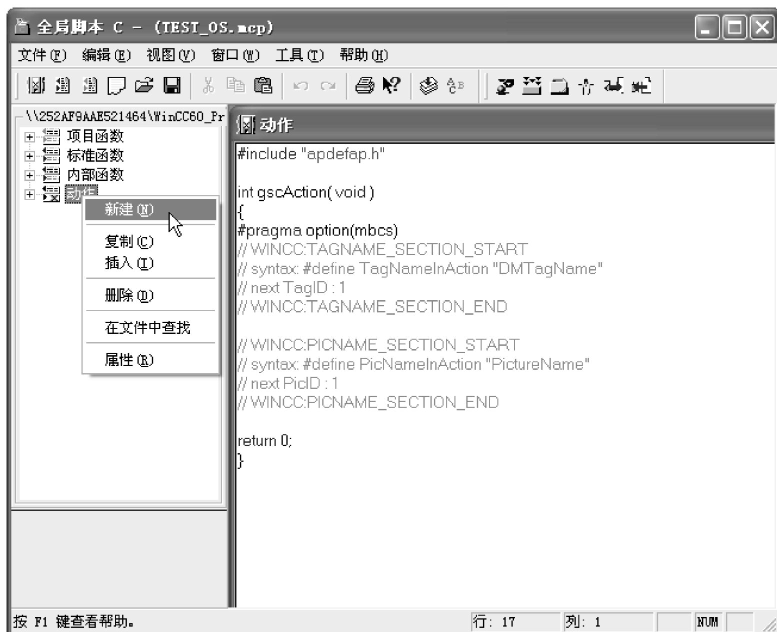



图 8-8　在 C-Script 全局脚本编辑器中新建一个全局动作

可以看到，新建的动作的基本框架已经生成。“#include“apdefap. h””用于向动作声明所有的项目函数、标准函数以及内部函数，也就是说有了该语句后，在新建的动作中可以调用所有的项目函数、标准函数以及内部函数。在自动生成的代码中包含两个注释块，若要使得 WinCC 的交叉索引编辑器可以访问该动作的内部信息，必须保留这两个注

释块。第一个注释块用于定义动作中所使用的 WinCC 变量，在下面的程序代码中必须使用在这里定义的变量名称，而不是实际的变量名称。第二个注释块用于定义动作中使用的 WinCC 画面，在下面的程序代码中也必须使用这里定义的画面名称，而不是实际的画面名称。

3）在编辑器窗口中输入动作代码，输入动作代码后的编辑窗口如图 8-9 所示。

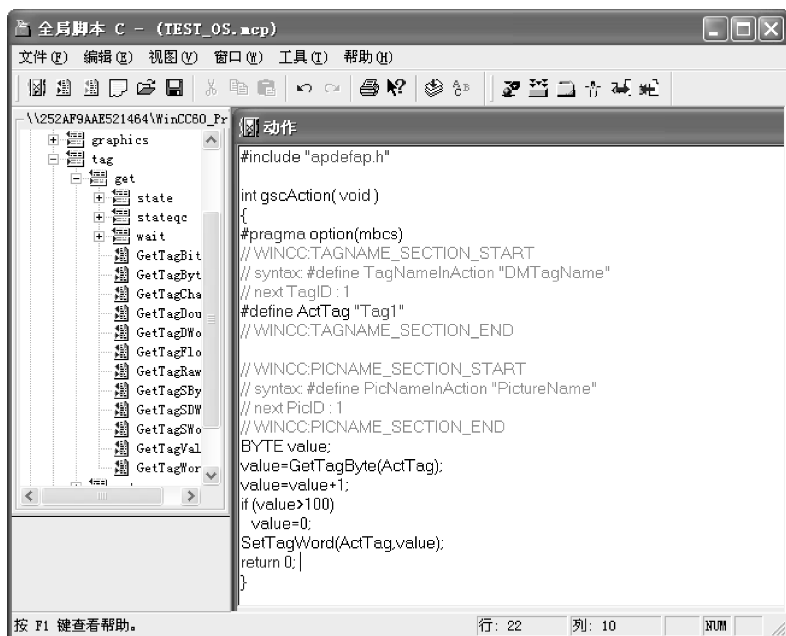



图 8-9　输入动作代码后的编辑窗口

可以看到，用“#define ActTag“Tag1””语句（宏定义语句）定义一个标识符“ActTag”表示 WinCC 变量 Tag1，后面均用该标识符代表变量 Tag1，这样 WinCC 的交叉索引编辑器可以访问变量 Tag1 的使用信息。

4）设置触发器。单击编辑工具栏上的按钮，打开“属性”对话框，选择“触发器”选项卡。在列表中选择“定时器”→“周期”，然后单击“添加”按钮，打开“添加触发器”对话框，如图 8-10 所示。在“事件”后的列表框中选择“标准周期”，在“周期”后的列表框中选择“1 秒”，设置触发器的名称为“counter”。

5）单击标准工具栏的按钮，保存新建的全局动作。

6）激活全局脚本运行系统。在 WinCC 项目管理器中打开“计算机属性”对话框，选择“启动”选项卡，勾选“全局脚本运行系统”前的复选框。

7）可以在过程画面中添加 1 个输入/输出域，并将该输入/输出域与变量 Tag1 建立变量连接，在运行系统中显示变量 Tag1 的值。

**5. 调试和诊断 ANSI－C 脚本**

WinCC 提供三个工具来分析 ANSI－C 脚本的运行状态，即应用程序窗口 GSC 运行系统、GSC 诊断和应用程序 apdiag. exe。

为了使用应用程序窗口 GSC 运行系统和 GSC 诊断，必须将其添加到过程画面中。该



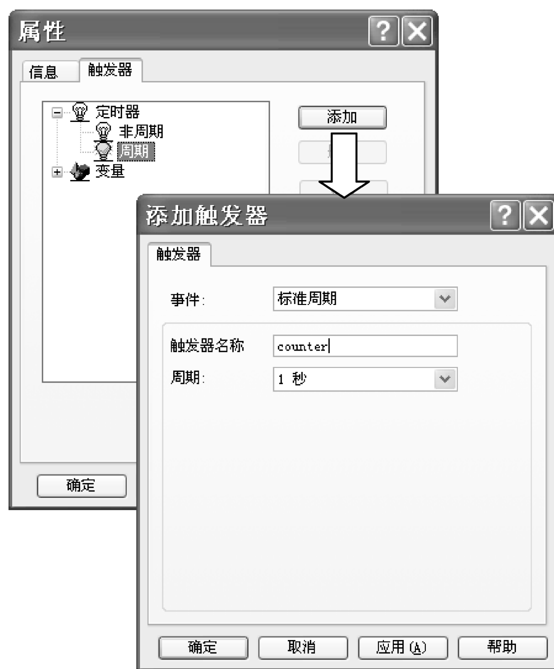


图 8-10　设置全局动作的触发器

过程画面可以是已存在的画面，也可以为诊断目的特别设置一个过程画面，然后在运行系统中调用该画面。

使用这些应用程序窗口，可实行下列不同的策略：

- 运行系统处于活动状态时，GSC 运行系统提供关于所有（全局脚本）动作的动态特性信息，支持单独启动和每个单个动作的开始和结束动作以及提供全局脚本编辑器入口点。
- GSC 诊断按调用的顺序输出 printf 指令（包含在动作中）的结果。这也适用于动作所调用的函数包含的 printf 指令。printf 指令可以用于输出变量的数值，也可以跟踪动作流和调用的函数。导致调用 OnErrorExecute 函数的错误条件，也将显示在 GSC 诊断窗口中。

（1）GSC 运行系统

GSC 运行系统是在运行系统中显示所有（全局脚本）动作的动态特性的窗口。另外，运行系统处于活动状态时，通过 GSC 运行系统，用户可影响单个动作的执行，并为全局脚本编辑器提供输入点。GSC 运行系统将输出下列信息：

- 动作：动作的名称。
- ID：动作的标识符。用于系统内部，例如在动作中发生一个错误，通过输出函数 OnErrorExecute，系统将输出动作的标识符和错误描述。GSC 运行系统用该标识符提供动作的名称。当运行系统终止或在运行系统处于活动状态的情况下保存动作时，标识符和动作名称之间的连接将不再有效。
- 状态：动作的当前状态，如图 8-11 所示。
- 激活时间间隔：两次调用动作之间的时间间隔，表现形式为“小时:分:秒”。

全局脚本 - 运行系统

| 动作 | ID | 状态 | 激活时间间隔 | 返回值 | 开始: | 下一次开始 | 出错消息: |
|---|---|---|---|---|---|---|---|
| gsccs1.pas | @c8 | 动作被激活。 | | 0 | 06/15/03 11:05:37 | | |
| gsccs1_1.pas | @cb | 动作被结束 | 0: 0: 2 | 0 | 06/15/03 11:07:51 | | |
| gsccs1_2.pas | @c9 | 动作被结束 | 0: 0: 2 | 0 | 06/15/03 11:07:51 | | |
| TankLevel.pas | @ca | 动作被结束 | 0: 0: 2 | 10 | 06/15/03 11:07:51 | | |

图 8-11　GSC 运行系统窗口

- 返回值：动作的返回值。
- 开始：当前启动动作的日期和时间。
- 下一次开始：再次启动动作的日期和时间。
- 出错消息：发生错误时包含错误文本。

将 GSC 运行系统添加到过程画面中的方法和步骤如下：

1）在图形编辑器的对象选项板中，选择“标准”→“智能对象”→“应用程序窗口”添加到过程画面中，将自动弹出“窗口内容”对话框，如图 8-12 所示。

2）在弹出的“窗口内容”对话框中选择“全局脚本”（Global Script），并单击“确定”按钮，确认条目并弹出“模板对话框”，如图 8-13 所示。




图 8-12　“窗口内容”对话框

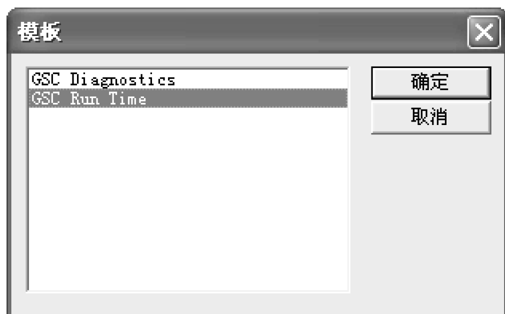



图 8-13　“模板”对话框

3）在“模板”对话框中，选择“GSC 运行系统”（GSC Run Time）。

4）单击“确定”按钮以确认选择。

（2）GSC 诊断

GSC 诊断窗口按调用的顺序输出 printf 指令（包含在动作中）的结果。为了使用 GSC 诊断，必须首先在过程画面中添加 GSC 诊断类型的应用程序窗口，其方法和步骤如下：

1）在图形编辑器的对象选项板中，选择“标准”→“智能对象”→“应用程序窗口”添加到过程画面中，将自动弹出“窗口内容”对话框，参考图 8-12。

2）在弹出的“窗口内容”对话框中选择“全局脚本”（Global Script），并单击“确定”按钮，确认条目并弹出“模板对话框”，参考图 8-13。

3）在“模板”对话框中，选择“GSC 诊断”（GSC Diagnostics）。

4）单击“确定”按钮以确认选择。

**6. 在函数或动作中使用动态链接库**

WinCC 允许用户使用自己的 DLL（Dynamic Link Library，动态链接库）。通过对各自的函数或动作进行必要补充，可以在函数和动作中启用现有 DLL 中的函数。在函数或动



作的起始处插入下列代码：

```
#pragma code("<Name>.dll")
 <Type of returned value> <Function_name 1>(...);
 <Type of returned value> <Function_name2>(...);
   …
 <Type of returned value> <Function_name n>(...);
#pragma code()
```

在函数或动作中插入上述代码后，来自“<Name.dll>”中的函数 <Function_name 1>…<Function_name n> 均已进行了声明，并可由各自的函数或动作进行调用，例如：

```
#pragma code("kernel32.dll")
  void GetLocalTime(LPSYSTEMTIME lpSystemTime);
#pragma code()
SYSTEMTIME st;
GetLocalTime(&st);
```

在 WinCC 中使用自己的 DLL 时，必须使用发布版。WinCC 是发布版，因而也使用系统 DLL 的发布版。如果在调试版中生成了自定义 DLL，则可能 DLL 的发布版和调试版二者都将装载，这样会增加需要的内存空间。

DLL 的结构必须使用 1 字节对齐方式进行设置。

注意：DLL 必须保存在“\ bin”目录中，或保存在“PATH”系统变量中所定义的路径中，该变量在操作系统属性中定义。

## 8.1.2 VBScript

WinCC 从 V6.0 开始集成 VBScript，既可以利用 VBScript 来使运行环境动态化，也可以用于创建动作和过程来动态化图形对象。VBScript 具有简单易学、便于调试等特点，深受广大工程技术人员的欢迎。

### 1. 过程、模块和动作

(1) 过程

过程是一段代码，类似于 C 语言中的函数，只需创建一次，就可以在项目中多次调用。这样可以省去很多重复性代码的编写。

相互关联的过程应该放在同一个模块中。在运行状态下，如果通过动作调用某个过程，则包含此过程的模块也会被加载。因此，应该注意以下两点：

- 当调用一幅画面时，加载的模块越多，运行状态下系统的性能越差。
- 模块越大，包含的过程越多，模块加载的时间越长。

因此，必须合理地组织模块，例如可以将用于特定系统或画面的过程组织在一个模块中；也可以按照功能构建模块，例如将具有计算功能的过程放在一个模块中。

在 WinCC 中，VBScript 过程具有以下特点：

- 过程可以由用户创建或修改。
- 过程可通过设置密码来保护过程代码。

- 过程不需要触发器。
- 过程存储于模块中。

WinCC 没有提供预定义过程，但是提供了代码模板和智能提示来简化编程。根据过程适用的范围不同，过程可以分为标准过程和项目过程：标准过程适用于计算机上的所有被创建的项目，而项目过程只适用于创建此过程的项目。

（2）模块

模块是一个文件，存放着一个或多个过程。在 WinCC 中，VBScript 模块具有以下特点：

- 可通过设置密码保护模块。
- 扩展名为“＊. bmo”。

模块根据存储在其中的过程的有效性不同而存在差异，分为标准模块和项目模块：

- 标准模块：包含所有项目可全局调用的过程，其存放路径为“WinCC installation directory\ApLib\ScriptLibStd\Module name. bmo”。
- 项目模块：包含某个项目可调用的过程，其存放路径为“Project directory >\ScriptLib\ Module name. bmo”。

项目模块存放在项目路径下，因此当项目被复制时，项目模块也被复制。

（3）动作

动作总是通过触发器来启动，也可以说是通过触发事件启动。例如，在运行下状态下，当单击画面上的某个对象时，定时时间到或某个变量被修改后，都可以触发动作。

动作在全局脚本中只定义一次，独立于画面而存在。全局脚本动作只在定义它的项目中有效。与画面对象相连接的动作，只在定义它的画面中有效。

动作具有以下特点：

- 动作由用户创建和修改。
- 全局脚本中的动作可以通过设置密码得到保护。
- 动作至少具有一个触发器。
- 全局脚本中的动作具有“＊. bac”文件扩展名。
- 全局脚本存放路径为“Project directory > \ ScriptAct \ Actionname. bmo”。

**2. 全局脚本 VBScript 编辑器**

在 WinCC 项目管理器的浏览窗口选择“全局脚本”→“VBS－Editor”，用鼠标右键单击并在弹出的菜单中选择“打开”命令，可以打开全局脚本 VBScript 编辑器，如图 8-14 所示。其中，浏览窗口用于管理项目模块、标准模块、动作和代码模板，选择相应的选项卡即可；编辑窗口用于输入和编辑脚本代码；输出窗口用于脚本编译后显示语法检查的结果等，在输出窗口双击相应的信息光标将定位到编辑窗口的对应代码处。

全局脚本 VBScript 编辑器的功能如下：

- 智能提示和语法高亮显示。
- 使用常规 VBS 函数。用鼠标右键单击编辑窗口的空白处，在弹出的菜单中选择“功能列表”命令，将显示常规的 VBS 函数。
- 使用对象、属性和方法列表。用鼠标右键单击编辑窗口的空白处，在弹出的菜单中选择“对象列表”或“属性/方法”，将显示对象、属性和方法。



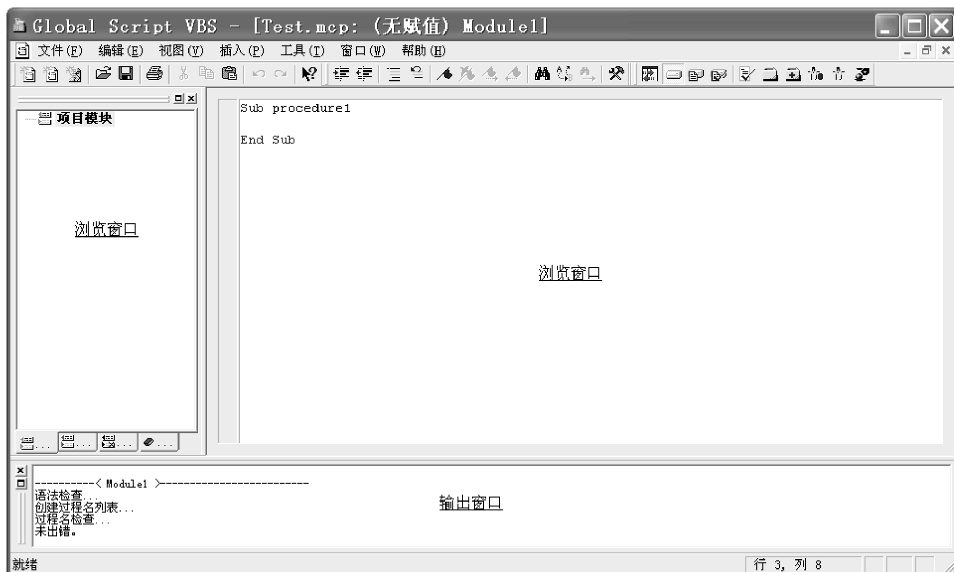



图 8-14　全局脚本 VBScript 编辑器

- 使用代码模板。单击浏览窗口的“代码模板”选项卡，将显示很多常用的代码，双击相应的代码模板，可以将其添加到编辑窗口。
- 使用选择对话框。在编辑器的工具栏单击按钮，可以打开“变量选择”对话框；单击按钮，可以打开“图形对象选择”对话框；单击按钮，可以打开“画面选择”对话框。
- 进行语法检查。在编辑器的工具栏单击按钮完成。

3. 创建和编辑过程

在全局脚本 VBScript 编辑器创建一个新过程时，WinCC 会自动地为该过程分配一个标准的过程名“procedure #”，其中“#”代表序号。可以在编辑窗口中修改过程名，以便动作能够调用该过程。当保存过程后，修改后的过程名将显示在浏览窗口中。过程名必须是唯一的，如果出现重名，将被认为是语法错误。

这里以创建求三个变量的平均值的过程为例，详细介绍在全局脚本 VBScript 编辑器中创建和编辑过程的方法和步骤。

（1）创建一个过程

1）在全局脚本 VBScript 编辑器中选择“项目模块”选项卡，在浏览窗口用鼠标右键单击“项目模块”，在弹出的菜单中选择“新建”→“项目模块”，一个没有返回值的过程将自动添加到编辑区。也可以打开一个已经存在的模块进行编辑：单击菜单“文件”→“打开”，在弹出的对话框中选择要打开的模块。

2）在现存的模块中插入一个过程。在浏览窗口中用鼠标右键单击需要插入过程的模块，在弹出的菜单中选择“添加新过程…”，将打开“新过程”对话框，如图 8-15 所示。输入过程名并选择是否带有返回的参数，然后一个变量声明和一个返回值就会输入到代码窗口中（选择带有返回的参数时，创建的过程的关键字为 Function…End Function；没有带返回的参数时，创建的过程的关键字为 Sub…End Sub）。








图 8-15　“新过程”对话框

（2）编写过程代码

在全局脚本 VBScript 编辑器的编辑窗口输入以下代码：

```
Function procedure2(tag1,tag2,tag3)
   Dim meanvalue
   meanvalue = (tag1 + tag2 + tag3)/3
   procedure2 = meanvalue
End Function
```

从上述程序代码可以看到：带有返回参数的过程，最后需要将返回值赋给过程名。

（3）语法检查

在全局脚本 VBScript 编辑器的工具栏单击 按钮，或在编辑窗口的空白处用鼠标右键单击，然后在弹出的菜单中选择“语法检查”，可以对输入的代码进行语法检查并根据输出窗口的输出信息确认是否出现语法错误。如果出现语法错误，可以在输出窗口双击相应的信息定位到出现错误的代码处，进行修改后再次进行语法检查，直到无语法错误为止。

**4. 创建和编辑动作**

VBS 动作主要用来使图形对象属性在运行时动态化，或者执行独立于画面的全局动作。VBS 动作分类如图 8-16 所示。

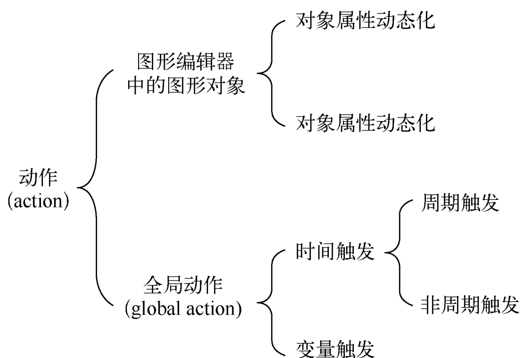



图 8-16　VBS 动作分类图

动作可以被分配给多个触发器。当某个触发事件发生时，动作就会被执行。需要注意以下事项：



- 全局脚本中的动作不能被同时执行，最后触发的动作排放在一个队列里，直到当前正在执行的动作完成之后才执行。
- 在图形编辑器中，循环触发和变量触发的动作不能同时启动。如果变量触发动作的执行阻碍了循环触发动作的执行，则当变量触发的动作完成之后，循环触发动作才执行。循环触发的动作在非执行周期被排放在一个队列里。目前执行的动作完成之后，循环触发的动作才以正常的周期执行。
- 在图形编辑器中，事件触发的动作不能同时执行。
- 不同类型的动作不会相互妨碍执行。全局脚本动作不会影响图形编辑器中的动作；同样，在图形编辑器中，周期循环触发/变量触发的动作不会影响事件触发的动作，如图 8-17 所示。

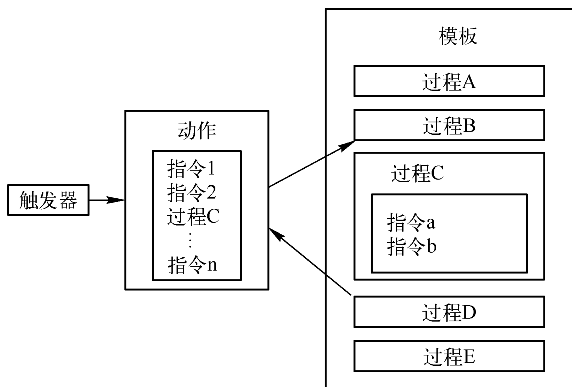



图 8-17　VBS 动作工作原理

创建和编辑全局 VBS 动作的方法和步骤如下：

1）在全局脚本 VBScript 编辑器的浏览窗口选择“动作”选项卡，在工具栏中单击“新建动作按钮，或单击菜单“文件”→“新建”→“动作”，将在编辑窗口新建一个动作。保存之后，新建的动作将显示在浏览窗口中。

2）编辑动作代码。动作代码的编辑方法与过程代码的编辑方法一样，可以利用编辑窗口提供的各种功能来进行编辑，例如代码模板、对象列表、属性/方法列表以及语法检测等。

3）添加动作的相关信息以及设置动作的触发器。在工具栏单击“信息/触发”按钮，打开全局动作的“属性”对话框，如图 8-18 所示。选择“信息”选项卡，可以添加动作的相关信息，例如为动作设置访问口令等。选择“触发器”选项卡，可以为动作设置相应的触发器，这与设置 ANSI－C 全局动作触发器的方法类似。




图 8-18　全局动作的“属性”对话框

4）单击工具栏的“保存”按钮或单击菜单“文件”→“保存”保存动作。

5）在 WinCC 项目管理器中打开“计算机属

性”对话框，选择“启动”选项卡，激活“全局脚本运行系统”。

5. 调试诊断 VBS 脚本

从 WinCC V6.0 开始就提供了一个 VBS 调试诊断工具来分析运行状态下 VBS 动作的执行情况，包括 GSC 运行和 GSC 诊断应用窗口以及 VBS 调试器。

GSC 运行和诊断应用窗口被用来添加到过程画面中，用法同 ANSI－C 脚本。不同的是，如果要在打印输出中间运算值到 GSC 诊断窗口中，VBS 的语法如下：

```
HMIRuntime.trace( <output> )
```

（1）调试器

在运行状态下，用来进行脚本调试的调试器有以下几种：

- Microsoft Script Debugger：包含在 WinCC 中（由于许可限制，Microsoft Script Debugger 不再包括在 WinCC V6.2 SP3 产品包内），能够在 Windows 2000 和 Windows XP 环境下应用。
- InterDev：包含在 Microsoft Visual Studio 的安装资源中。
- Microsoft Script Editor（MSE）Debugger：包含在 Microsoft Office 中。

要使用 Microsoft Script Debugger，必须先进行安装，然后在“计算机属性”对话框中进行激活。在 WinCC 安装光盘的安装菜单中选择“附加软件”，然后在“附加软件”窗口中用鼠标单击“Microsoft Script Debugger”，将开始安装过程。在 WinCC 项目管理器中打开“计算机属性”对话框，选择“运行系统”选项卡，可以根据项目的需求选择激活全局脚本系统和图形运行系统的调试器，如图 8-19 所示。




图 8-19　激活脚本调试器



选择“启动排错程序”复选框后，如果在运行状态下脚本出现错误，调试器将直接启动；选择“显示出错”对话框，如果在运行状态下发生脚本错误，调试器不会直接启动，而是显示一个错误对话框，其中包含错误信息，调试器可以单击对话框中的“确认”按钮启动。

脚本调试器具有以下功能：

- 观看需要调试的脚本源代码。
- 脚本的单步运行检查。
- 显示变量和属性的修改值。
- 监控脚本执行过程。

（2）VBS 调试器

VBS 调试器如图 8-20 所示。

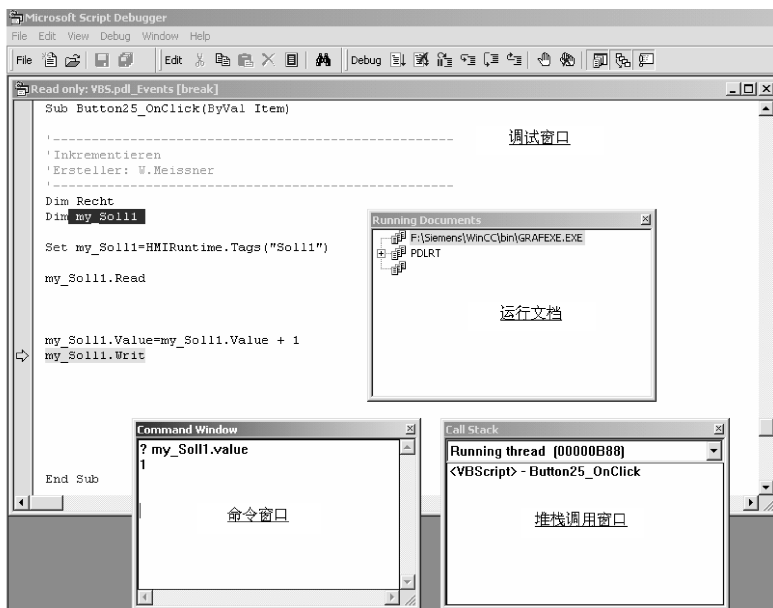



图 8-20　VBS 调试器

1）命令窗口（Command Window）。命令窗口可通过菜单“Veiw”→“Command Window”菜单项打开。可以在命令窗口中进行以下操作：

- 输入命令。可以直接输入命令，并在脚本中直接执行。
- 更改变量的值。变量值可以直接在命令窗口中编译和修改，包括脚本中的变量和全局变量。
- 修改属性。可以在命令窗口中读/写目前脚本中所有对象的属性。

2）运行文档窗口（Running Documents Window）。运行文档窗口可以通过菜单“View”→“Running Documents”菜单项打开。运行文档窗口显示在 WinCC 运行状态下，所有运行的脚本根据类型不同而归属不同的分支，即分为全局脚本（Global Script Runtime）和图形运行系统脚本（PDLRT）。图形运行系统脚本根据触发条件不同，又分为触发控制脚本（picturename_trigger）和事件控制脚本（picturename_events）。

3）堆栈调用窗口（Call Stack Window）。堆栈调用窗口可通过菜单“View”→“Call

Stack”菜单项打开。此窗口主要用来显示所有运行的动作和调用的过程。当一个过程被调用时，它的名字被添加到窗口列表中。当过程调用结束后，过程的名称从窗口列表中消失。

（3）动作过程在调试器中的名称

在调试器中，动作和过程的名称不同于存储在 WinCC 脚本中的名字，它们遵循表 8-2 中的规则。

表 8-2　脚本编辑器中动作和过程的名称

| 动 作 类 型 | 脚本文件名称 |
|---|---|
| 属性上的循环或变量触发事件 | ObjectName_PropertyName_Trigger |
| 鼠标事件 | ObjectName_OnClick |
| | ObiectName_OnLButtonDown |
| | ObjectName_OnLButtonUp |
| | ObjectName_OnRButtonDown |
| | ObjectName_OnRButtonUp |
| 键盘事件 | ObjectName_OnKeyDown |
| | ObjectName_OnKeyUp |
| 对象事件 | ObjectName_OnObjectChanged |
| | ObjectName_OnSetFocus |
| 属性事件 | ObjectName_PropertyName_OnPropertyChanged |
| | ObjectName_—PropertyName_OnPropertyStatcChanged |
| 画面事件 | Document_OnOpen |
| | Document_OnClosed |

（4）调试脚本的过程

1）选择脚本。

- 从 Windows 启动菜单中启动脚本调试器：“开始”→“程序”→“附件”→“Microsoft Script Debugger”。
- 在调试器中，单击菜单“View”→“Running Documents”，从运行文档窗口中选择要调试的脚本。
- 双击运行文档窗口中需要调试的脚本文件，脚本文件就会在调试窗口中打开（写保护）。

2）设置断点。断点经常设置在代码中容易出错的地方。

- 将光标放在需要设置断点的位置。
- 单击菜单命令“Debug”→“Toggle Breakpoint”，下一行要执行的代码就会标记一个红点。
- 切换 WinCC 到运行状态，触发动作使脚本运行，调试器停留在第一个断点处，并用黄色高亮度显示。

3）单步运行。按〈F8〉快捷键单步运行脚本文件，也可以使用菜单命令“Debug”→“Step into”或“Debug”→“Step over”进行调试。

4）确定或修改变量或属性值。

- 脚本文件中至少设置一个断点。



- 切换 WinCC 到运行状态下，触发动作，执行脚本。调试器将停在第一个断点处。
- 单击菜单“View”→“Command Window”，打开命令窗口。
- 为了确定变量或属性的值，先输入一个字符“?”，然后输入空格以及变量或属性的名字，例如“? mytag”按〈Enter〉键，执行命令。
- 如果要修改变量/属性，则用 VBS 的赋值语法。

## 8.1.3 应用实例

本节通过几个实例来进一步介绍全局脚本的概念以及组态方法。

**1. 调用项目函数在固定位置显示对话框——ANSI-C 函数**

如图 8-21 所示，在运行状态下，单击名称为“激活对话框”的按钮，将在画面的固定位置显示对话框（画面窗口），从对话框中可以监控电机的运行状态。该实例的实现首先要在全局脚本 ANSI-C 编辑器中组态一个项目函数 Box_mit_Pos1，然后在图形编辑器中添加按钮对象和画面窗口对象，并在按钮的鼠标事件中组态调用项目函数 Box_mit_Pos1 的 C 动作。

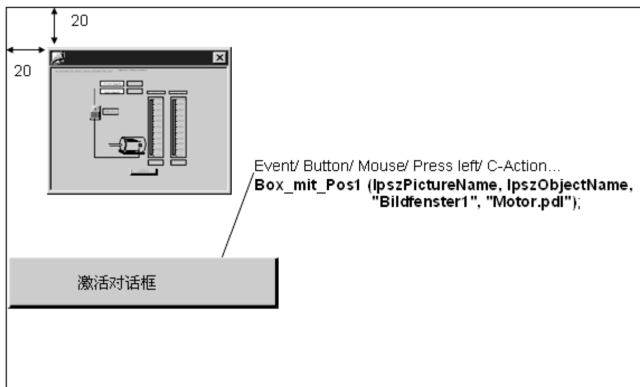



图 8-21 在按钮的鼠标动作中调用项目函数显示对话框

具体组态方法和步骤如下：

（1）组态项目函数 Box_mit_Pos1

1）在 WinCC 项目管理器的浏览窗口单击“全局脚本”，将显示“C-Editor”和“VBS-Editor”，双击“C-Editor”或用鼠标右键单击“C-Editor”，然后在弹出的菜单中选择“打开”命令，打开 C－Script 全局脚本编辑器。

2）在 C-Script 全局脚本编辑器的浏览窗口用鼠标右键单击“项目函数”，在弹出的菜单中选择“新建”命令，打开“编辑窗口”，同时在编辑窗口中新建一个名称为“New_Function”的项目函数，将新建的项目函数的名称修改为 Box_mit_Pos1，并编写如下函数代码：

```
void Box_mit_Pos1(char * lpszPictureName, char * lpszObjectName, char * lpszBScreenName,
            char * lpszBoxName)
{
    int BScreenXPos, BScreenYPos;
    BScreenXPos = 20;
    BScreenYPos = 20;
```

```
    SetLeft( lpszPictureName, lpszBScreenName, BScreenXPos);
    SetTop( lpszPictureName, lpszBScreenName, BScreenYPos);
    SetPictureName( lpszPictureName, lpszBScreenName, lpszBoxName);
    SetVisible( lpszPictureName, lpszBScreenName, 1);
}
```

3）编译保存函数。单击编辑工具栏中的按钮，对新建的函数进行编译，检查“输出”窗口输出的来自编译器的消息，如果编译器报告出错，则必须更正函数代码，完成后再次进行编译查错，直到编译正确为止。

4）单击标准工具栏的按钮，保存新建的项目函数。

项目函数 Box_mit_Pos1 包含 4 个形参，其含义如下：

- lpszPictureName：当前屏幕的名称。
- lpszObjectName：操作对象的名称。
- lpszBScreenName：画面窗口对象的名称。
- lpszBoxName：在画面窗口中输出的画面的名称。

在项目函数中调用了 ANSI－C 脚本的 4 个内部函数 SetLeft、SetTop、SetPictureName 和 SetVisible，其中 SetLeft 和 SetTop 用于指定画面窗口对象在画面中的位置，而 SetPictureName 用于指定画面窗口输出的画面名称，SetVisible 用于设定画面窗口的显示/隐藏状态。

（2）组态画面

1）在 WinCC 图形编辑器中新建一个画面，从对象选项板的“标准”→“智能对象”列中选择 1 个“画面窗口”添加到画面中。用鼠标右键单击添加的画面窗口对象，在弹出的菜单中选择“属性”命令，打开“对象属性”窗口。选择“属性”选项卡，在左侧窗口中选中“画面窗口”，在右侧窗口中将属性“对象名称”的静态值设置为“PictureWindow1”。在左侧窗口中选中“其它”，在右侧窗口中将属性“显示”的静态值设置为“否”。

2）从对象选项板的“标准”→“窗口对象”列中选择 1 个“按钮”添加到画面中。在自动弹出的“按钮组态”对话框中将“文本”设置为“激活对话框”，然后单击“确定”退出对话框。用鼠标右键单击添加的按钮，在弹出的菜单中选择“属性”命令，打开“对象属性”窗口。选择“事件”选项卡，在左侧窗口中选中“鼠标”，在右侧窗口中用鼠标右键单击“按左键”行的白色闪电，在弹出的菜单中选择“C 动作…”，打开“编辑动作”窗口，在代码窗口中输入以下代码：

```
void OnLButtonDown(char * lpszPictureName, char * lpszObjectName, char * lpszPropertyName,
                   UINT nFlags, int x, int y)
{
    Box_mit_Pos1(lpszPictureName, lpszObjectName, "PictureWindow1","Motor.pdl");
}
```

3）单击工具栏中的按钮，对新建的 C 动作进行编译，直到没有错误时，按“确定”按钮退出“编辑动作”窗口。

4）在图形编辑器中的工具栏中，单击“保存”按钮保存组态的画面。

在 C 动作中调用项目函数 Box_mit_Pos1 时对应的传输了 4 个参数，其中参数“lpsz-



PictureName”指定了步骤 1）中新建的画面，参数“lpszObjectName”指定了步骤 2）创建的按钮，“PictureWindow1”即步骤 1）创建的画面窗口，“Motor. pdl”是画面窗口输出的画面名称，该画面必须在之前已经组态好。

（3）运行 WinCC 项目，测试组态的结果。

如果在按钮鼠标事件的 C 动作代码中调用项目函数 Box_mit_Pos2，则能够更灵活地控制对话框在画面中显示的位置。项目函数 Box_mit_Pos2 的代码如下，运行结果示意图如图 8-22 所示，可以看到对话框在按钮的右下角显示。

```
void Box_mit_Pos2(char * lpszPictureName, char * lpszObjectName,     char * lpszBScreenr-
                  Name, char * lpszBoxName)
{
//变量声明
    int ButtonXPos, ButtonYPos, ButtonWidth, ButtonHeight;
    int ScreenWidth, ScreenHeight;
    int BScreenXPos, BScreenYPos, BScreenWidth, BScreenHeight;
    int XInterval = 0, YInterval = 0;
    //取按钮在画面中的位置和尺寸
    ButtonXPos = GetLeft(lpszPictureName,lpszObjectName);
    ButtonYPos = GetTop(lpszPictureName,lpszObjectName);
    ButtonWidth = GetWidth(lpszPictureName,lpszObjectName);
    ButtonHeight = GetHeight(lpszPictureName,lpszObjectName);
    //取画面的尺寸
    ScreenWidth = GetWidth(lpszPictureName,NULL);
    ScreenHeight = GetHeight(lpszPictureName,NULL);
    //取画面窗口的尺寸
    BScreenWidth = GetWidth(lpszPictureName,lpszBWindowName);
    BWindowHeight = GetHeight(lpszPictureName,lpszBWindowName);
    //指定画面窗口的输出位置(按钮的右下角)
    BScreenXPos = ButtonXPos + ButtonWidth + XInterval;
    BWindowYPos = ButtonYPos + ButtonHeight + YInterval;
    if ((BScreenXPos + BScreenWidth) > ScreenWidth)
    {
      BScreenXPos = ButtonXPos - BScreenWidth - XInterval;
    }
    if ((BScreenYPos + BScreenHeight) > ScreenHeight)
    {
        BScreenYPos = ScreenHeight - BScreenHeight;
    }
    SetLeft(lpszPictureName,lpszBScreenName,BScreenXPos);
    SetTop(lpszPictureName,lpszBScreenName,BScreenYPos);
    SetPictureName(lpszPictureName,lpszBScreenName,lpszBoxName);
    SetVisible(lpszPictureName,lpszBScreenName,1);
}
```



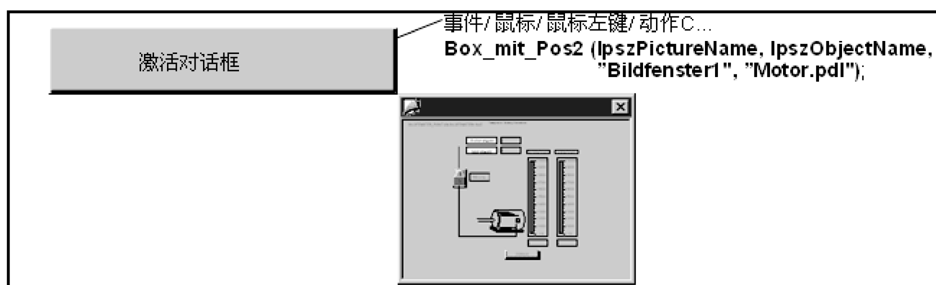



图 8-22　调用项目函数 Box_mit_Pos2 指定对话框的位置

实际上，可以通过调用脚本程序将对话框放在用户希望的画面任意位置。为了建立操作对象（即图 8-21 和图 8-22 中的按钮）的参考，如果对话框出现在操作对象被使用的位置是很有好处的，可以通过下面的步骤来获得操作对象的位置并指定对话框的位置：

- 扫描操作对象和整个画面的属性。
- 扫描对话窗口的尺寸。
- 指定对话窗口的外观位置（临近操作对象的右边或左边）。

### 2. 计算系统运行时间——ANSI－C 全局动作

WinCC 中的项目动作系统（Project Action System，PAS）可以进行独立于画面的动作。这些动作在后台运行，可以循环（时间触发）和非循环（在一个特定的时间）触发，甚至可以依靠一个事件（变量）启动。这些动作的执行是适合于整个项目的。单个的动作在全局脚本中产生。

这里以计算系统运行时间的全局动作为例，在运行系统中当电机驱动器或其他部件工作时，内部计数器将自动增加计数，并将分钟和小时记录下来。以 1 分钟为周期循环触发计数动作，记录操作的小时数，并将该小时数保存在分配的变量中。

组态方法和步骤如下：

1）在 WinCC 项目管理器的变量管理中创建类型为“无符号 16 位数”的内部变量 Minutes 和 OperatingHours，类型为“二进制变量”的内部变量 Z_On。

2）在 C-Script 全局脚本编辑器中创建一个动作，动作的文件名为“OperatingHours.PAS”，触发器设置为标准周期 1 分钟，动作代码如下：

```
int gscAction( void)
{
    WORDHOUR, MINUTE;
    BOOLZ_ON;
    Z_ON = GetTagBit ( "Z_On" ) ;
    if (Z_ON)
    {
        MINUTE = GetTagWord ( " Minutes" ) ;
        MINUTE = MINUTE + 1;
        SetTagWord ( " Minutes" , MINUTE) ;
        HOUR = MINUTE/60;
```



```
            SetTagWord("operatingHours", HOUR);
        }
        return(HOUR);
    }
```

3）在 WinCC 项目管理器中，打开“计算机属性”对话框，选择“启动”选项卡，勾选“全局脚本运行系统”前的复选框。

4）在画面中添加 1 个输入/输出域，并将其与变量 OperatingHours 建立变量连接。

5）运行项目，测试组态结果。

在动作代码中，WinCC 变量 Z_On 的值决定是否进行计数（if 语句进行状态判定），该变量可以用于反映对象（例如电机驱动器）的运行状态，当对象运行时将其置 1，全局动作中计数器将增加计数，当对象为停止状态时将其置 0，计数器将不计数。

#### 3. 在 WinCC 运行系统中启动外部应用程序——VBS 动作

在 WinCC 运行系统中，单击画面中的按钮，启动其他应用程序（例如 Windows 应用程序记事本）。组态方法是在图形编辑器中添加 1 个按钮对象到画面中，用 VBS 动作组态按钮的鼠标按左键事件，动作代码如下：

```
Sub OnLButtonDown(ByVal Item, ByVal Flags, ByVal x, ByVal y)
  Dim objWshShell
  Set objWshShell = CreateObject("Wscript.Shell")
  objWshShell.Run "Notepad Example.txt", 1
End Sub
```

#### 4. ActiveX 控件的调用——VBS 动作

下面几个例子分别演示了如何调用嵌入到 WinCC 画面中的 ActiveX 控件的属性和方法。

（1）WinCC FunctionTrend Control（WinCC 函数趋势控件）

以下代码实现了在对象名为“Control1”的 WinCC 函数趋势控件中输出抛物线：

```
Dim lngFactor
Dim dblAxisX
Dim dblAxisY
Dim objTrendControl
Set objTrendControl = ScreenItems("Control1")
For lngFactor = -100 To 100
  dblAxisX = CDbl(lngFactor * 0.02)
  dblAxisY = CDbl(dblAxisX * dblAxisX + 2 * dblAxisX + 1)
  objTrendControl.DataX = dblAxisX
  objTrendControl.DataY = dblAxisY
  objTrendControl.InsertData = True
Next
```

（2）ComboBox 组合框

以下代码实现了为对象名为“ComboBox1”的组合框添加列表项：

```
Dim cboComboBox
Set cboComboBox = ScreenItems("ComboBox1")
cboComboBox. AddItem"1_ComboBox_Field"
cboComboBox. AddItem"2_ComboBox_Field"
cboComboBox. AddItem"3_ComboBox_Field"
cboComboBox. FontBold = True
cboComboBox. FontItalic = True
cboComboBox. Index = 2
```

（3）控制 MS Web Brower

以下代码实现了对添加到 WinCC 画面的名称为“WebControl”的 Web 浏览器对象的控制：

```
Dim objWebBrowser
Set objWebBrowser = ScreenItems("WebControl")
objWebBrowser. Navigate "http://www. ad. siemens. com. cn"
…
objWebBrowser. GoBack
…
objWebBrowser. GoForward
…
objWebBrowser. Refresh
…
objWebBrowser. GoHome
…
objWebBrowser. GoSearch
…
objWebBrowser. Stop
…
```

## 8.2　WinCC 的开放性接口

从一开始，SIMATIC WinCC 就代表了最高水平的开放性和集成性，因为 WinCC 始终以 Microsoft 技术作为后盾。

### 8.2.1　WinCC 的开放性接口概貌

WinCC 几乎集成了 Microsoft 所有的开放性接口技术，包括 ActiveX、DDE、OPC、VBA、VBS、OLE、API 以及 Microsoft 强大而高效的数据库 Microsoft SQL Server 等，如图 8-23 所示。通过 Microsoft 的这些开放性接口，可以将一些应用程序集成到 WinCC 系统中，使得 WinCC 的扩展应用更加方便简单。



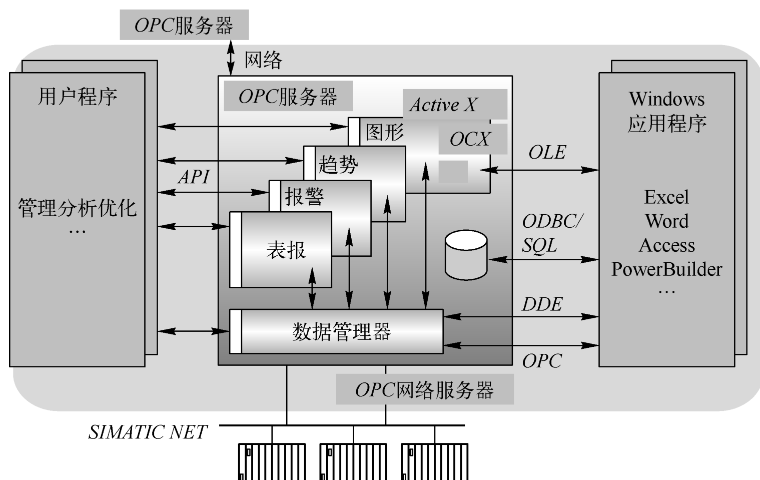



图 8-23　WinCC 的开放性接口示意图

### 1. ActiveX 控件

ActiveX 是自身具有用户接口的程序模块的 Windows 标准。这些程序模块被称作 ActiveX 控件。例如，ActiveX 控件可包含特殊按钮或图形显示元素。WinCC 提供了大量的 ActiveX 控件，附加的 ActiveX 控件可从其他供应商处获取或单独编程。使用 Visual Basic 创建单个 ActiveX 控件时，WinCC 的选件 IndustrialX 提供支持。要确保 WinCC 功能正确运作，应在使用之前详细地测试这些控件。

使用拖放方法可将 ActiveX 控件集成到用户的 WinCC 画面中。

### 2. OLE

OLE 是“Object Linking and Embedding”（对象链接和嵌入）的缩写，并且是 Microsoft Windows 应用程序之间进行数据交换的标准。它将来自一个应用程序的数据插入到用户自己的应用程序中。一个典型的实例就是将图表插入到文本中，随后通过双击图表对其进行编辑。Windows 将打开原始图形程序以允许进行编辑。

应用 OLE 技术可以在 WinCC 项目中将 Excel 表格集成到 WinCC 画面中，并使用表格中的数据作为配方数据。

### 3. OPC

OPC 是“OLE for Process Control（用于过程控制的 OLE）”的缩写，是为自动化技术特别开发的一种 OLE 接口。使用该接口，任意 OPC 激活的组件可相互通信。用户在组态期间不必考虑接口的具体细节。

WinCC 可以是 OPC 客户机或 OPC 服务器。在作为 OPC 客户机时，WinCC 将访问其他应用程序的数据。当 WinCC 用做 OPC 服务器时，WinCC 数据将可供其他应用程序使用。WinCC 允许进行以下类型的访问：

- 通过 WinCC OPC DA 服务器访问 WinCC 变量。
- 通过 WinCC OPC HDA 服务器访问 WinCC 归档系统。
- 通过 WinCC OPC A&E 服务器访问 WinCC 消息系统。

### 4. SQL

SQL 可用来访问 WinCC 数据库的内容。SQL 是“Structured Query Language（结构化

查询语言)”的缩写，是一种用于访问数据库的标准化语言。

利用 SQL 查询到的数据既可以用于其他应用程序，也可以导入到其他数据库中。

5. API

WinCC 的每一个组件都有一个 API（Application Program Interface，应用程序接口）用以开放 WinCC 供其他应用程序访问。因此，单个应用程序可以影响 WinCC，可以访问组态运行系统数据或对过程进行干预。ODK 选件（Open Development Kit，开放式开发工具）包含该接口的文档和大量实例。

6. ANSI-C

对于 WinCC 项目中过程的动态，WinCC 支持函数和动作的使用。这些函数和动作以 ANSI - C 编写。

7. VBS

在 WinCC 中，除 C-Script 以外，程序语言 VBScript 也可作为应用程序接口。VBScript（VBS）提供运行时图形运行系统的变量和对象访问，并允许独立画面动作的执行。除了指定的 WinCC 应用程序之外，也可使用 VBS 常规功能来访问 Windows 环境。

8. VBA

VBA（Visual Basic for Application）接口是自定义 WinCC 的另一个选择。在图形编辑器中，组态时可以用 VBA 自动频繁循环工作步骤。此外，可以利用支持 VBA 的 Microsoft Office 家族产品。

9. ADO/OLE DB

可以通过 ADO/OLE DB 接口访问 WinCC 归档数据库。

本章将通过一些实例介绍这些技术的应用。OPC 技术的概念和应用将在第 9 章进行详细介绍。

## 8.2.2　OLE 接口及其应用

利用 OLE 技术使得其他应用程序可以访问 WinCC 的数据，本节阐述 Microsoft Office Excel 和 WinCC 通过 OLE Automation 2.0 实现互连，WinCC 变量的值可以在 Excel 的单元格中显示，还可以将 Excel 单元格的值写给 WinCC 变量。

1. 组态 WinCC

在 WinCC 项目管理器的变量管理中创建 2 个类型为“无符号 16 位数”的内部变量 VAR_WORD_1 和 VAR_WORD_2。

在 WinCC 图形编辑器中新建一个画面，画面名称为“WinCC_OLE_Excel. pdl”。在画面中添加 2 个输入/输出域，在输入/输出域的“属性”对话框中将其“输出格式”属性组态为“999999”，“域类型”属性组态为“输入/输出域”，并将 2 个输入/输出域分别与变量 VAR_WORD_1 和 VAR_WORD_2 建立变量连接。

在画面中添加 1 个按钮，按钮显示的文本为“WinCC_OLE_Excel. xls”，打开按钮的“属性”对话框，在“事件”选项卡中用 VBS 动作组态按钮的“鼠标”→“按左键”事件，动作代码如下：

```
Sub OnLButtonDown(ByVal Item, ByVal Flags, ByVal x, ByVal y)
    Dim objExcelApp
```



```
    Set objExcelApp = CreateObject("Excel. Application")
    objExcelApp. Visible = True
    objExcelApp. Workbooks. Open "f:\WinCC_OLE_Excel. xls"
End Sub
```

在 WinCC 运行状态下单击该按钮，将打开应用程序 Excel，同时打开 Excel 文件“WinCC_OLE_Excel. xls”，动作代码中“f:\WinCC_OLE_Excel. xls”标识了该文件的存放路径。

激活 WinCC 后，切换到画面“WinCC_OLE_Excel. pdl”，运行结果如图 8-24 所示。

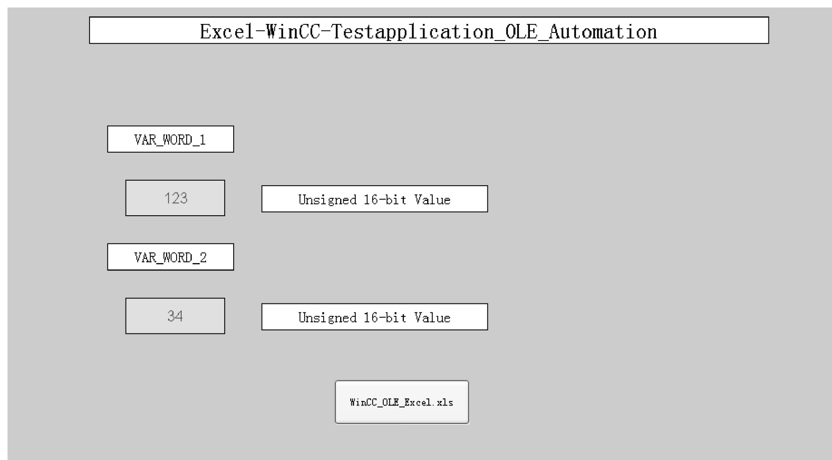



图 8-24　WinCC 组态的运行结果

### 2. 组态 Excel

在 Excel 中主要是组态从 WinCC 中写或读 WinCC 变量 VAR_WORD_1 和 VAR_WORD_2 的值。

单击 Excel 的菜单“工具”→“宏（M）”→“宏（M）…”，可以打开“宏”对话框，如图 8-25 所示。在“宏名”框中输入宏的名称，然后单击“创建”按钮可以创建 1 个宏，选中创建的宏，然后单击“编辑”按钮，可以打开“Microsoft Visual Basic”，输入并编辑宏的程序。按此方法创建 2 个宏：GetValue 和 SetValue。




图 8-25　“宏”对话框

宏 GetValue 的程序代码如下：

```
Sub GetValue( )
      Dim mcp        As Object
      Dim value     As Variant
      Dim vars       As Variant
      Dim cell            As Variant
      Dim i                  As Integer
      vars = Array( "VAR_WORD_1" , "VAR_WORD_2" )
      Set mcp = CreateObject( "WinCC – Runtime – Project" )
      For i =0 To 1
            cell = "D"  & (i  +  7)
            value = mcp. GetValue( vars( i) )
            Range( cell). Select
            ActiveCell. FormulaR1C1 = Str( value)
      Next
End Sub
```

宏 SetValue 的程序代码如下：

```
Sub GetValue( )
      Dim mcp        As Object
      Dim value     As Variant
      Dim vars       As Variant
      Dim cell            As Variant
      Dim i                  As Integer
      vars = Array( "VAR_WORD_1" , "VAR_WORD_2" )
      Set mcp = CreateObject( "WinCC – Runtime – Project" )
      For i =0 To 1
            cell = "D"  & (i +7)
            value = mcp. GetValue( vars( i) )
            Range( cell). Select
            ActiveCell. FormulaR1C1 = Str( value)
      Next
End Sub
```

在 Excel 中，单击菜单“视图”→“工具栏”→“控件工具箱”，可以打开控件工具箱，在控件工具箱中选中“命令按钮”，鼠标将变为十字型，此时可以在 Excel 的表格中添加命令按钮。按此方法在 Excel 表格中添加 2 个按钮，按钮显示文本（对应的按钮属性为 Caption）分别为“GetValue( )”和“SetValue( )”。

用鼠标双击文本为“GetValue( )”的按钮，为该按钮的鼠标单击事件添加以下代码：

```
Private Sub CommandButton1_Click( )
    Call GetValue
End Sub
```



同样的方法为文本为“SetValue( )”的按钮添加以下代码：

```
Private Sub CommandButton2_Click( )
    Call SetValue
End Sub
```

在 Excel 的工具栏单击“保存”按钮或单击菜单“文件”→“保存”，将组态的 Excel 文件保存为“f:\WinCC_OLE_Excel. xls”，退出 Excel 应用程序。

在图 8-24 所示的 WinCC 画面中单击文本为“WinCC_OLE_Excel. xls”的按钮，在弹出的对话框中选择“启用宏”，将打开 Excel 应用程序，同时打开 Excel 文件“WinCC_OLE_Excel. xls”，如图 8-26 所示。

Microsoft Excel - WinCC_OLE_Excel

文件(F) 编辑(E) 视图(V) 插入(I) 格式(O) 工具(T) 数据(D) 窗口(W) WinCC 帮助(H) Adobe PDF

I18

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | WinCC和Excel的数据交换 | | | | | |
| 2–3 | OLE测试应用 | | | | | |
| 4 | | | | | | |
| 5–6 | | | | GetValue() | SetValue() | |
| 7 | | VAR_WORD_1 | | 123 | 75 | |
| 8 | | VAR_WORD_2 | | 34 | 63 | |
| 9 | | | | | | |

图 8-26　Excel 文件运行结果

在 Excel 表格中单击“GetValue( )”按钮时，WinCC 变量 VAR_WORD_1 和 VAR_WORD_2 的值分别被读到 Excel 表格的单元格 D7 和 D8 中。当单击“SetValue( )”开关时，单元格 E7 和 E8 的数值分别被写进 WinCC 变量 VAR_WORD_1 和 VAR_WORD_2 中。

## 8.2.3　DDE 接口及应用

本节以 WinCC 与 Microsoft Office Excel 的本地 DDE 通信为例进行介绍。

### 1. WinCC 作为 DDE 服务器，Excel 作为 DDE 客户端

组态方法和步骤如下：

1）在 WinCC 项目管理器中，打开“计算机属性”对话框，选择“启动”选项卡，单击“添加（A）…”按钮，打开“添加应用程序”对话框，如图 8-27 所示。单击“应用程序（A）”框旁边的“浏览（B）…”按钮，添加 WinCC 安装目录下的应用程序 DdeServ. exe，同时在“命令行参数（C）”框中添加命令行参数“/v/n”。

2）运行 WinCC 项目后，将弹出“WinCC DDE－Server”对话框，如图 8-28 所示。单击“Tag list”按钮，打开“变量－项目”对话框，在对话框中选择 Excel 要访问的变量。单击“WinCC DDE－Server”对话框中的“确定”按钮，则该变量的连接信息将复制到 Windows 的剪贴板。

3）打开 Excel，双击选中任意一个单元格，将 Windows 剪贴板的内容粘贴到该单元格中，然后在第一个字符前添加“＝”号，完成对单元格的输入，如图 8-29 所示。








图 8-27　“添加应用程序”对话框

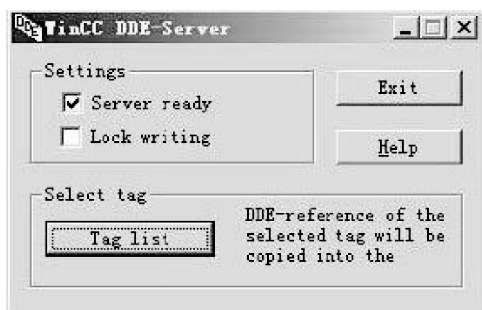



图 8-28　“WinCC DDE－Server”对话框

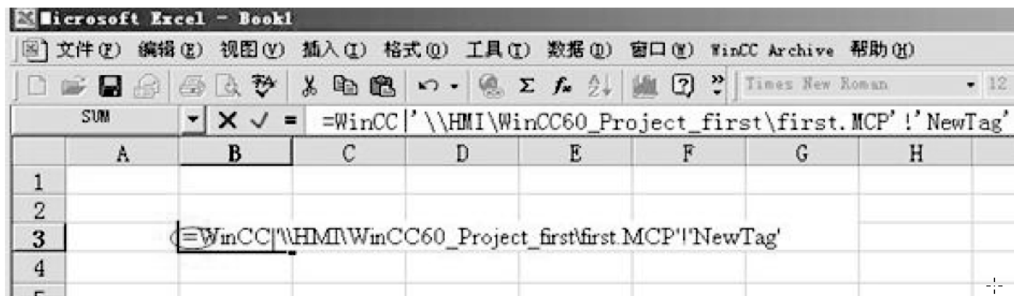



图 8-29　在 Excel 的单元格中输入连接 WinCC 变量的信息

4）如果需要访问多个变量，可重复步骤 2）和 3）。

此时，Excel 的单元格可以显示 WinCC 变量的值了。



需要注意以下两点：

- 应用程序 DdeServ. exe 要一直保持运行状态。
- WinCC 与 Excel 的这种 DDE 连接是单向的，即 WinCC 变量值的改变可以在 Excel 的单元格中体现，但是无法在 Excel 的单元格中修改 WinCC 变量的值。

### 2. WinCC 作为 DDE 客户端，Excel 作为 DDE 服务器

组态方法和步骤如下：

1）在 WinCC 项目管理器中用鼠标右键单击“变量管理”，在弹出的菜单中选择“添加新的驱动程序”，选择“Windows DDE. chn”创建 DDE 通信通道。

2）用鼠标右键单击创建的 DDE 通道，在弹出的菜单中选择“新驱动程序的连接”，建立 DDE 通道下的连接“WiCC_DDE_Excel”，连接的属性设置如图 8-30 所示。“计算机名称”框为空，“应用程序”框中输入“Excel”，在“主题”框中输入 Excel 文件的文件名和工作表名称“［Excel 文件名］工作表名”，例如“［WinCC_DDE_Excel. xls］Sheet1”。

3）在新建的连接“WiCC_DDE_Excel”中创建新的变量，在“变量属性”对话框中单击“选择”按钮，打开该变量的属性对话框，如图 8-31 所示。在“条目名称”框中输入：r 行数 c 列数，例如 r1c1 表示 Excel 工作表中第一行第一列的单元格。单击“确定”按钮退出对话框，完成一个变量的组态。

4）如果需要创建多个变量，重复步骤 3）。

5）在 WinCC 的画面中添加输入/输出域，并将其与上面创建的变量建立连接。

6）运行 WinCC 项目，打开 Excel 文件，测试组态结果。Excel 工作表中可以显示 WinCC 变量的值，在对应单元格输入数值，可以将该数值写到 WinCC 变量中。




图 8-30　DDE 通信连接的属性




图 8-31　新建 DDE 通信的变量

## 8.2.4　API 接口及应用

每个 WinCC 应用程序，例如变量管理器、图形编辑器、变量记录、报警记录、全局脚本等，都有一个 API 接口，用以开放 WinCC 供其他应用程序访问，当然也可以通过 WinCC 使用 Windows API。

其他应用程序通过 WinCC API 访问 WinCC 应用程序的原理如图 8-32 所示。要创建访问 WinCC 的外部应用程序，主要需要 3 个部件：

- 外部的 C 语言编译器（例如 Microsoft Visual C ++6.0）。
- WinCC API（ *.LIB ）库文件。
- WinCC API（ *.H ）头文件。

为了能在外部使用 WinCC API，ODK（开放式开发工具包）是很必要的。ODK 包含 WinCC API 的库文件和头文件，使得一些必要的部件在使用 WinCC API 时可用，这样用户就可以生成自己的对象甚至整个 WinCC 选项。

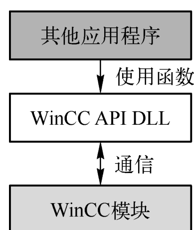



图 8-32　应用程序访问 WinCC 组件的原理

CDK（Channel Development Kit，通道开发工具包）用于通道 DLL 开发（ 第三方驱动程序接口）。

WinCC 的动作编程隐含使用 WinCC API。另外，动态向导使用 WinCC API 进行工作。以下介绍几个 WinCC API 方面的应用实例。

### 1. 调用 DLL（API）函数的实例

在 WinCC 图形编辑器中，从对象选项板选择“标准”→“窗口对象”→“按钮”添加到画面中，按钮显示的文本为“用 DLL 函数计算”，打开按钮的“属性”窗口，用 C



动作组态按钮的“鼠标”→“按左键”事件，动作代码如下：

```
void OnLButtonDown (char * lpszPictureName, char * lpszObjectName, char * lpszPropertyName,
                    UINT nFlags, int x, int y)
{
  #pragma code("c:\a_WinCC_Kurs_Prj_301_00\ab_pas\library\demo_dll.dll")
     int CountingDll(int start, int end);
  #pragma code()
  int result, start, end, partial step;
  partial step = GetTagDWord("partial step");
  printf("\r\n\r\noutput of 10 partial steps with DLL function ");
  printf("to max. value: %d",partial step * 10);
  for (start = 0;start < 10;start ++)
  {
     end = start * partial step + partial step;
     result = countingDll(start * partial step,end);
     printf("\r\nReturn value DLL function: (%d) %d",start,result);
  }
}
```

在上面的代码中，处理了一个从开始值到结束值的循环，中间结果被输出到诊断窗口中。通过直接比较发现，除非操作较多，否则很容易发现使用 DLL 函数没有多少时间优势。

程序段#pragma code("c:\a_WinCC_Kurs_Prj_301_00\ab_pas\library\demo_dll.dll")...#pragma code()是标准的调用 API 函数的编写方法,"c:\a_WinCC_Kurs_Prj_301_00 \ab_pas\library\demo_dll.dll"给出了 API 函数的存放路径。

来自 WinCC 的动作编程提供了多数调用 DLL 函数的可能性，并且有可能访问保存在 DLL 中的外部函数。这些 DLL 既可以是标准的 DLL，也可以是用户自己创建的。特别推荐在需要大量计算的地方调用 DLL 的函数。因为这些代码已经以可执行的方式保存在 DLL 中，不必在运行中进行解释，所以对大量的计算有更快的处理速度。

用户要创建 DLL，可以使用 Visual C++ 中的应用向导，例如“MFC - AppWizard (dll)”。将要用到的所有函数都被输入到这个 DLL 中，并且被声明为外部的“C”。另外，每个函数都必须输入到输出表格中。

**2. 触发计算机蜂鸣器的声音**

用 C 动作组态按钮的“鼠标”→“按左键”事件，动作代码如下：

```
void OnLButtonDown (char * lpszPictureName, char * lpszObjectName, char * lpszPropertyName,
                    UINT nFlags, int x, int y)
{
  #pragma code("kernel32.dll");
    BOOL Beep(DWORD dwFreq,DWORD dwDuration);
  #pragma code();
  Beep(500,500);
}
```

API 函数“Beep”的参数“dwFreq”定义了蜂鸣器声音的频率，参数“dwDuration”定义了蜂鸣器声音的持续时间。

3. 调用 API 函数 Sleep( )实现等待功能

下面的程序描述了如何调用 API 函数 Sleep( )实现等待(“睡眠”) 功能。

```
#pragma code( "Kernel32. dll" )
  void Sleep( intMilliseconds) ;
#pragma code( )
Sleep(1000) ; //以毫秒为单位定义等待时间
```

执行 Sleep( )函数后，在定义的等待时间内，C 脚本程序的处理将暂停，无法响应中断函数的中断请求，因此应该慎重使用 Sleep( )函数。

### 8.2.5　使用 ActiveX 控件

OCX 和 ActiveX 对象提供了未被 WinCC 的对象默认包含的某些功能。WinCC 使 OCX 和 ActiveX 对象可以嵌入到画面中，从而实现更多复杂的功能。

ActiveX 是基于 COM（Component Object Model，组件对象模型）的可视化控件结构的名称，是一种封装技术，提供封装 COM 组件并将其置入应用程序的一种方法。在 Windows 系统中注册的所有 ActiveX 控件均可用于 WinCC。

1. 在 WinCC 中直接插入 ActiveX 控件

图形编辑器对象选项板中的“控件”选项卡中包含各种控件，这些控件可以直接插入到画面中。在“控件”选项卡中用鼠标右键单击“选择”，在弹出的菜单中选择条目“添加/删除”，将打开“选择 OCX 控件”对话框，如图 8-33 所示。在“可用的 OCX (数字)”区域中，“选择 OCX 控件”对话框显示操作系统中记录的所有 ActiveX 控件。在读入注册信息之后，确切数字将显示在该区域的标题中。红色复选标记表示对象选项板“控件”选项卡对话框中的可用控件。所选择的 ActiveX 控件的路径和程序标识号显示在“详细资料”区域中。单击“注册 OCX”按钮可以添加 OCX 控件，选中某个 OCX 控件后单击“取消注册 OCX”，则从列表中删除该控件。

在对象选项板中插入可用的 ActiveX 控件后，就可以采用拖放的方式将其插入到 WinCC 画面中。

2. 用 VBScript 访问 ActiveX 控件

本章 8.1.3 节举例说明了如何用 VBScript 访问 ActiveX 控件，一般采用的方法是用 VBScript 中的 ScreenItems 对象来访问并修改 ActiveX 控件的属性。在画面中插入 ActiveX 控件后，会有一个对象名，假定为“Control1”，则可以通过以下代码访问并修改它的属性：

```
Dim objControl
Set objControl1 = ScreenItems(“Contrl1”)
objControl. Height = 5
objControl. Width = 10
…
```






图 8-33 “选择 OCX 控件”对话框

### 8.2.6 ADO/OLE－DB 访问归档数据库

#### 1. ADO/OLE－DB 的概念

OLE－DB 是一种快速访问不同数据的开放性标准，与 ODBC 标准不同。ODBC 是建立在 Windows API 函数基础上的，只能通过它访问关系型数据库。而 OLE－DB 是建立在 COM 和 DCOM 基础上的，可以访问关系型数据库或非关系型数据库。

简单地说，OLE－DB 提供一种统一的数据访问接口，使得数据的使用者（应用程序）可以使用同样的方法访问各种数据，而不用考虑数据的具体存储地点、格式或类型。OLE－DB 标准的具体实现是一组 C++ API 函数，就像 ODBC 标准中的 ODBC－API 一样，不同的是，OLE－DB 的 API 是符合 COM 标准、基于对象的（ODBC－API 则是简单的 C－API）。OLE DB 与 ODBC 标准都是为了提供统一的访问数据接口，但 ODBC 标准的对象是基于 SQL 的数据源，而 OLE DB 的对象则是范围更为广泛的任何数据存储。从这个意义上说，符合 ODBC 标准的数据源是符合 OLE－DB 标准的数据存储的子集。符合 ODBC 标准的数据源要符合 OLE－DB 标准，还必须提供相应的 OLE－DB 服务程序（Service Provider），就像 SQL Server 要符合 ODBC 标准，必须提供 SQL Server ODBC 驱动程序一样。现在，微软已经为所有的 ODBC 数据源提供了一个统一的 OLE DB 服务程序，即 Microsoft OLE DB/ODBC。OLE－DB 标准的 API 是 C++ API，只能供 C++ 语言调用。为了使得流行的各种编程语言都可以编写符合 OLE DB 标准的应用程序，微软在 OLE DB API 之上，提供了一种面向对象、与语言无关的应用编程接口，这就是 ActiveX Data Objects，即 ADO。

OLE－DB 层和数据库的连接是通过一个数据库提供者（provider）而建立的。OLE－DB 接口和提供者是由不同的制造商提供的。

除了 WinCC OLE－DB 接口之外，还可以通过 Microsoft OLE－DB 和 ODBC 来访问 WinCC 的归档数据。但是，通过 WinCC OLE DB Provider，可以直接访问 WinCC 压缩归档

数据，而且是透明访问；而通过 Microsoft OLE DB 或 ODBC 只能访问 WinCC 非压缩归档数据和消息归档数据。

2. WinCC 选件 WinCC/Connectivity Pack

从 WinCC V6.2 开始，所有的历史数据通过算法进行了压缩处理，无法通过标准的 SQL 语句访问，需要在安装 WinCC 选件 WinCC/Connectivity Pack 的基础上，根据 WinCC/Connectivity Pack 的语法编写程序访问历史归档值和历史报警信息。

WinCC/Connectivity Pack 软件包包含 4 个主要的接口组件：OPC HDA Server、OPC A&E Server、OPC XML DA Server 以及 WinCC OLE – DB 接口。利用 WinCC Connectivity Pack，可以通过 WinCC OLE – DB 或 OPC HDA 访问 WinCC 的归档数据，也可以通过 OPC XML 跨操作系统平台访问 WinCC 的当前值。同样，还可以利用 OPC A&E 把 WinCC 的消息传递给其他系统。

因此，当使用 WinCC OLE – DB 访问 WinCC 归档数据库时，如果是本地访问，本地计算机除了需要安装 WinCC 基本系统外，还需安装 WinCC 选件 WinCC/Connectivity Pack；如果是远程访问，提供归档数据的服务器除需安装 WinCC 基本系统外，还需安装 WinCC 选件 WinCC/Connectivity Pack。

3. 利用 ADO/WinCC OLE – DB 访问数据库的语法

（1）与数据库建立连接的语法

连接字符串的结构是:" Provider = Name of the OLE DB Provider; Catalog = Database Name; Data Source = Server Name"，各个参数的含义见表 8-3。

表 8-3　与数据库建立连接的语法

| 参　　数 | 描　　述 |
|---|---|
| provider | OLE DB provider 的名称：WinCCOLEDBProvider |
| catalog | WinCC 数据库的名称：<br>对于 WinCC 运行数据库，用数据库的名称末尾加一个“R”：<DatabaseName_R>.<br>如果是用 WinCC Archive Connector 连接交换出的 WinCC 归档，就是用其符号名称 |
| Data source | 服务器名称：<br>如果是本地访问：\ WinCC<br>如果是远程访问：ComputerName \ WinCC |

例如：

```
Set conn = CreateObject("ADODB. Connection")
conn. open"Provider = WinCCOLEDBProvider. 1; Catalog = CC_OpenArch_03_05_27_14_11_46R;
                  Data Source = . \WinCC"
```

（2）查询过程值归档的语法

查询过程值归档的语法如下：

```
TAG:R, <ValueID or ValueName>, <TimeBegin>, <TimeEnd>
```

查询过程值归档的时间可以采用绝对时间和相对时间，采用绝对时间各个参数的含义见表 8-4，采用相对时间时各个参数的含义见表 8-5。



表 8-4　采用绝对时间为查询条件时查询过程值的语法

| 参　　数 | 描　　述 |
|---|---|
| ValueID | 数据库表中的 ValueID |
| ValueName | “ArchiveName \ ValueName” 格式的 ValueName 值，ValueName 必须用单引号 |
| TimeBegin | 起始时间格式：YYYY - MM - DD hh. mm. ss. mmm |
| TimeEnd | 终止时间格式：YYYY - MM - DD hh. mm. ss. mmm |

表 8-5　采用相对时间为查询条件时查询过程值的语法

| 参　　数 | 描　　述 |
|---|---|
| ValueID | 数据库表中的 ValueID |
| ValueName | “ArchiveName \ ValueName” 格式的 ValueName 值，ValueName 必须用单引号 |
| TimeBegin | 0000 - 00 - 00 00:00:00. 000：从记录的开始读起 |
| TimeEnd | 0000 - 00 - 00 00:00:00. 000：读到记录的结束 |
| 例 1 | <TimeBegin> = From 2002 - 02 - 02 12:00:00. 000 until <TimeEnd> = 0000 - 00 - 00 00:00:10. 000：向前读 10 秒 |
| 例 2 | <TimeBegin> = From 0000 - 00 - 00 00:00:10. 000 until <TimeEnd> = 2002 - 02 - 02 12:00:00. 000：向后读 10 秒 |

（3）查询报警信息归档的语法

查询报警信息归档的语法如下：

```
ALARMVIEW:SELECT * FROM <ViewName>[WHERE <condition>..., optional]
```

各个参数的含义见表 8-6。

表 8-6　查询报警信息归档的语法

| 参　　数 | 描　　述 |
|---|---|
| ViewName | 数据库表的名称。数据表由期望的语言来指定<br>AlgViewDeu：德语消息归档数据<br>AlgViewEnu：英语消息归档数据<br>AlgViewEsp：西班牙语消息归档数据<br>AlgViewFra：法语消息归档数据<br>AlgViewIta：意大利语消息归档数据 |
| Condition | 过滤条件，e. g. ：<br>DateTime >'2003 - 06 - 01' AND DateTime <'2003 - 07 - 01'<br>DateTime >'2003 - 06 - 01 17：30：00'<br>MsgNr = 5<br>MsgNr in（4，5）<br>State = 2<br>用时间过滤，只能用绝对时间 |

### 4. ADO/WinCC OLE - DB 数据库访问应用实例

（1）用 ADO/WinCC OLE - DB 访问过程值归档

在本应用实例中，WinCC 变量最后 10 分钟的值从 WinCC 运行数据库中读出，并显示在一个 ListView 控件中，输出值限制在 1000 以内。具体组态方法和步骤如下：

1）在 WinCC 项目管理器的变量管理中创建类型为“无符号 16 位数”的变量 Tag1。

2）在 WinCC 变量记录中创建一个过程值归档 ProcessValueArchive1，将 Tag1 添加到该过程值归档中。

3）创建一个 Visual Basic 6.0 的项目，在窗体中添加 1 个 MS Windows Common Control 6.0“Listview Control”，对象名为 ListView1，ListView1 中的列由脚本创建。在窗体中添加 1 个命令按钮，按钮显示的文本为“read data”，将下列程序代码复制到该命令按钮的单击（Click）事件中：

```
Dim sPro As String
Dim sDsn As String
Dim sSer As String
Dim sCon As String
Dim sSql As String
Dim conn As Object
Dim oRs As Object
Dim oCom As Object
Dim oItem As Object
Dim m, n
' 为 ADODB 创建连接字符串
sPro = "Provider = WinCCOLEDBProvider.1;"
sDsn = "Catalog = CC_A1479_05_12_29_11_13_04R;"
sSer = "Data Source = .\WinCC"
sCon = sPro + sDsn + sSer
' 在 sSql 中定义命令文本(相对时间)
sSql = "Tag:R,8','0000 -00 -02 00:5:00.000','0000 -00 -00 00:00:00.000'"
' 也可以是 sSql = "Tag:R','ProcessValueArchive1\OPC_Excel','0000 -00 -02 00:05:00.000',
            '0000 -00 -00 00:00:00.000'"
' 建立和数据库的连接
Set conn = CreateObject("ADODB.Connection")
conn.ConnectionString = sCon
conn.CursorLocation = 3
conn.Open
' 使用命令文本进行查询
Set oRs = CreateObject("ADODB.Recordset")
Set oCom = CreateObject("ADODB.Command")
' adCmdText    =1
oCom.CommandType = 1
Set oCom.ActiveConnection = conn
oCom.CommandText = sSql
' 填充记录集到 ListView1 中
```



```
Set oRs = oCom. Execute
m = oRs. Fields. Count
ListView1. ColumnHeaders. Clear
ListView1. ColumnHeaders. Add , , CStr( oRs. Fields(1). Name), 140
ListView1. ColumnHeaders. Add , , CStr( oRs. Fields(2). Name), 70
ListView1. ColumnHeaders. Add , , CStr( oRs. Fields(3). Name), 70
ListView1. ListItems. Clear
If (m >0) Then
    oRs. MoveFirst
    n =0
    Do While Not oRs. EOF
        n = n + 1
        Set oItem = ListView1. ListItems. Add( )
        oItem. Text = Left( CStr( oRs. Fields(1). Value), 23)
        oItem. SubItems(1) = FormatNumber( oRs. Fields(2). Value, 4)
        oItem. SubItems(2) = Hex( oRs. Fields(3). Value)
      ' If (n >10000) Then Exit Do
        oRs. MoveNext
    Loop
    oRs. Close
Else

End If
Set oRs = Nothing
conn. Close
Set conn = Nothing
```

4）运行 WinCC 项目，启动 Visual Basic6. 0 的项目，单击窗体上的“read data”按钮测试组态结果。

注意：用户需要将程序代码“sDsn = " Catalog = CC_A1479_05_12_29_11_13_04R;" ”中的 WinCC 运行数据库名称“CC_A1479_05_12_29_11_13_04R”更改为自己的 WinCC 运行数据库名称，数据库名称可在 WinCC 处于运行状态时，在 SQL Server Management Studio 中查看。

（2）用 ADO/WinCC OLE – DB 访问消息系统归档

在本实例中，从消息系统归档数据中读取 10 分钟时间间隔的数据。数据带有时间标记、消息编号、状态和消息类型显示在 ListView 对象中。具体组态方法和步骤如下：

1）在 WinCC 报警记录中组态消息系统，并在“计算机属性”对话框中激活报警记录。

2）创建一个 Visual Basic 6. 0 的项目，在窗体中添加 1 个 MS Windows Common Control 6. 0“Listview Control”，对象名为 ListView2，ListView2 中的列由脚本创建。在窗体中添加 1 个命令按钮，按钮显示的文本为“read alarm”，将下列程序代码复制到该命令按钮的单击（Click）事件中：

```
Dim sPro As String
Dim sDsn As String
Dim sSer As String
Dim sCon As String
Dim sSql As String
Dim conn As Object
Dim oRs As Object
Dim oCom As Object
Dim oItem As Object
Dim m, n
' 为 ADODB 创建连接字符串
sPro = "Provider = WinCCOLEDBProvider. 1;"
sDsn = "Catalog = CC_A1479_05_12_29_11_13_04R;"
sSer = "Data Source = . \WinCC"
sCon = sPro + sDsn + sSer
' 在 sSql 中定义命令文本(相对时间)
sSql = "ALARMVIEW:Select * FROM AlgViewEnu WHERE DateTime > 2005 - 10 - 02 00:5:
    00.000 and DateTime < 2005 - 12 - 30 00:00:00.000 "
' 创建连接
Set conn = CreateObject("ADODB. Connection")
conn. ConnectionString = sCon
conn. CursorLocation = 3
conn. Open
' 使用命令文本进行查询
Set oRs = CreateObject("ADODB. Recordset")
Set oCom = CreateObject("ADODB. Command")
' adCmdText  = 1
oCom. CommandType = 1
Set oCom. ActiveConnection = conn
oCom. CommandText = sSql
' 填充记录集到 ListView2 中
Set oRs = oCom. Execute
m = oRs. Fields. Count
ListView2. ColumnHeaders. Clear
ListView2. ColumnHeaders. Add , , CStr(oRs. Fields(2). Name), 140
ListView2. ColumnHeaders. Add , , CStr(oRs. Fields(0). Name), 60
ListView2. ColumnHeaders. Add , , CStr(oRs. Fields(1). Name), 60
ListView2. ColumnHeaders. Add , , CStr(oRs. Fields(37). Name), 100
ListView2. ListItems. Clear
If (m > 0) Then
    oRs. MoveFirst
    n = 0
    Do While Not oRs. EOF
```



```
            n = n + 1
            Set oItem = ListView2. ListItems. Add( )
            oItem. Text = CStr( oRs. Fields(2). Value)
            oItem. SubItems(1) = CStr( oRs. Fields(0). Value)
            oItem. SubItems(2) = CStr( oRs. Fields(1). Value)
            oItem. SubItems(3) = CStr( oRs. Fields(37). Value)
          ' If (n > 10000) Then Exit Do
            oRs. MoveNext
        Loop
        oRs. Close
    Else
    End If
    Set oRs = Nothing
    conn. Close
    Set conn = Nothing
```

3）运行 WinCC 项目，启动 Visual Basic6. 0 的项目，单击窗体上的“read alarm”按钮测试组态结果。

注意：用户需要将程序代码“sDsn = " Catalog = CC_A1479_05_12_29_11_13_04R;" ”中的 WinCC 运行数据库名称“CC_A1479_05_12_29_11_13_04R”更改为自己的 WinCC 运行数据库名称，数据库名称可在 WinCC 处于运行状态时，在 SQL Server Management Studio 中查看。

## 8.3 组态技巧与技术问答

### 8.3.1 在 WinCC 中读取系统时间的方法

用 C 动作组态如按钮的“鼠标”→“按左键”事件，动作代码如下：

```
void OnLButtonDown (char * lpszPictureName, char * lpszObjectName, char * lpszPropertyName,
                    UINT nFlags, int x, int y)
{
        #pragma code( " kernel32. dll" ) ;
            Void GetLocalTimes( SYSTEMTIME * lpst) ;
        #pragma code( ) ;
        SYSTEMTIME time;
        GetLocalTime( &time) ;
        SetTagWord( " TagYear" , time. wYear) ;
        SetTagWord( " TagMonth" , time. wMonth) ;
        SetTagWord( " TagWeek" , time. wDayOfWeek) ;
        SetTagWord( " TagDay" , time. wDay) ;
        SetTagWord( " TagHorus" , time. wHour) ;
        SetTagWord( " TagMinute" , time. wMinute) ;
```

```
        SetTagWord("TagSecond",time.wSecond);
        SetTagWord("TagMillisecond",time.wMilliseconds);
    }
```

动作代码中，TagYear、TagMonth、TagWeek 等是“无符号 16 位数”类型的 WinCC 内部变量，调用了 WinCC API 函数 GetLocalTimes 来获取系统的时间。

### 8.3.2　用 C 脚本解决授权点数不够用问题的方法

在采用 WinCC 设计 HMI 系统的过程中，经常会遇到项目中需要组态的工艺参数超过 WinCC 版本限制的授权点数（即过程变量数目）的问题。这可以通过 C 脚本编程实现多个工艺参数打包成一个过程变量传送来解决。例如某个配料称重系统有 146 个参数，超过了 WinCC 的 128 个授权点数的限制，就可以用 C 语言编程解决这一问题而不需要购买更多的授权。

基本思想就是将多个参数在下位控制器的内存中进行连续排列，然后在 WinCC 中定义一个过程变量，其长度是多个参数之和，取得这个过程变量的值后，通过 C 脚本程序将其分成多个参数。

例如，下位控制器（例如 PLC）有两个参数 LTN44001 和 LTN44023，类型均为 16 位的整数，分别存放在 DD99.DW146 和 DD99.DW148 中，在 WinCC 的变量管理器中创建一个过程变量，变量名为 PackageTag，类型为“有符号 32 位数”，其地址为 DD99.DBD146，如图 8-34所示。再定义两个类型为“有符号 16 位数”的内部变量 LTN44001 和 LTN44023。




图 8-34　创建传送变量 PackageTag

在全局脚本 ANSI-C 编辑器中组态全局动作，动作的触发器为变量 PackageTag 的变量触发，更新周期为“有变化时”，动作代码如下：



```
int gscAction( void )
{
   #pragma option(mbcs)
   union
   {
      long int tagDword;
      int tagWord[2];
   }union1;
   union1.tagDword = GetTagSDword("PackageTag");
   SetTagSWord("LTN44001",union1.tagWord[0]);
   SetTagSWord("LTN44023",union1.tagWord[1]);
   return 0;
}
```

在动作代码中用关键字 union 定义了一个共用体类型，包含两个成员：变量 tagDword 和数组tagWord[2]，由共用体的概念可知变量 tagDword 和数组 tagWord[2]共用系统内存（4 个字节），因此数组元素 tagWord[0]与变量 tagDword 的低2 字节共享内存，tagWord[1]与变量 tagDword 的高 2 字节共享内存。由此达到了将一个过程变量 PackageTag 分成两个参数的目的。

### 8.3.3 用 C 脚本实现用户登录日志

用户登录日志主要包括用户的登录和注销信息，以便查询用户使用 HMI 系统的时间段。对于一个成熟的 HMI 系统来说，生成和保存用户登录日志非常重要，当系统出现故障或发生事故时，有利于认定事故的责任，提升系统的管理水平。

在 WinCC 中，可以通过 C 脚本编程来实现生成用户登录日志。在全局脚本 ANSI－C 编辑器中组态全局动作，动作代码如下：

```
#include "apdefap.h"
int gscAction( void )
{//花括号 1 开始
      #pragma code("kernel32.dll")
         void GetLocalTime(LPSYSTEMTIME lpSystemTime);
      #pragma code()
      char * username;
      char buf[128];
      static char preuser[128];
      unsigned a,b,c,d,e,f;
      FILE * fp;
      SYSTEMTIME sysTime;
      //读取系统时间,并且复制给变量 a,b,c,d,e,f
      GetLocalTime(&sysTime);
      a = sysTime.wHour;
      b = sysTime.wMinute;
```

```
        c = sysTime. wSecond;
        f = sysTime. wYear;
        e = sysTime. wMonth;
        d = sysTime. wDay;
        //得到当前用户名称
        username = GetTagChar("@ CurrentUser");
        fp = fopen("c:\\wincclog. txt", "a + ");
        if(strcmp(username, preuser)! =0) //如果当前用户名称和前一个用户名不同
        {//花括号 2 开始
            if((strcmp(username, "")! =0)&&(strcmp(preuser, "")==0)) //如果当前用户
                  名称不空同时前一个用户名为空
        {
          sprintf(buf, "用户:%s\t 登陆时间是:\t %d - %d - %d,%d - %d - %d\n",
          username,a,b,c,d,e,f);
          fputs(buf, fp);
        }
        else
        {//花括号 3 开始
          if((strcmp(username, "")==0)&&(strcmp(preuser, "")! =0)) //如果当前用户名
                称为空同时前一个用户名不空
        {
          sprintf(buf, "用户:%s\t 退出时间是:\t%d - %d - %d,%d - %d - %d\n", preuser,
                a,b,c,d,e,f);
          fputs(buf, fp);
        }
        else
        {
          sprintf(buf, "用户:%s\t 退出时间是:\t%d - %d - %d,%d - %d - %d\n", preuser,
                a,b,c,d,e,f);
          fputs(buf, fp);
          sprintf(buf, "用户:%s\t 登陆时间是:\t%d - %d - %d,%d - %d - %d\n", user-
                name,a,b,c,d,e,f);
          fputs(buf, fp);
        }
      }//花括号 3 结束
    }//花括号 2 结束
        strcpy(preuser, username);
        fclose(fp);
        return 0;
  }//花括号 1 结束
```

上述全局动作的触发器为变量@ CurrentUser 的变量触发，更新周期为有变化时。

# 第9章

# 基于 OPC 的数据访问

OPC 是 OLE for Process Control（用于过程控制的 OLE）的缩写，它是微软公司的对象链接和嵌入技术（Object Linking and Embedding，OLE）在过程控制方面的应用。OPC 定义了一套适用于过程控制应用，支持过程数据访问、报警、事件与历史数据访问等的功能接口。OPC 协议在工业自动化控制领域的应用已经十分广泛。

WinCC 作为西门子公司的组态软件，完全支持 OPC 协议，既可用做 OPC 客户端，也可以充当 OPC 服务器。本章将详细介绍 OPC 的基本概念、WinCC 在 OPC 方面的应用以及 WinCC 组态 OPC 服务器和 OPC 客户端的方法。

## 9.1 OPC 的基本概念

OPC（用于过程控制的 OLE）是一个标准的、与制造商无关的软件接口。OPC 接口的基础是 Microsoft Windows 的 COM（Component Object Model，组件对象模型）和 DCOM（Distributed Component Object Model，分布式组件对象模型）技术。

OPC 技术的实现由两部分组成：OPC 服务器和 OPC 客户端。OPC 服务器完成的工作就是收集现场设备的数据信息，然后通过标准的 OPC 接口传送给 OPC 客户端应用。OPC 客户端则通过标准的 OPC 接口接收数据信息。在具体的实现过程中，用户可以根据自己的需要挑选相应的规范来使用。

OLE 是 COM 的前身，其最初含义是指在程序之间链接和嵌入对象数据，为用户提供建立混合文档的手段，例如将 Excel 表格插入到 Powerpoint 幻灯片中。但最初的 OLE 规范并未获得大多数软件开发商的支持，直到包括 COM 对象的新的 OLE 规范的提出，OLE 技术才逐渐获得开发商的认同。

COM 是位于同一计算机上的对象之间通信的标准协议，它提供了接口和内部组件通信标准的编程模型，其最大的特点就是采用“接口”来实现客户端和服务器之间的标准通信。所谓的 COM 接口是指由 COM 对象提供的执行某些特殊服务的函数和程序的集合。COM 为一个应用（客户端）提供了调用其他应用（服务器）服务的标准方式，服务器将它的服务用对象来实现，客户端可以通过标准的方法访问这些对象。

DCOM 是 COM 功能的扩展，在不同计算机之间的信息传递可以使用 DCOM。

### 9.1.1 OPC 的优点

OPC 具有以下优点：

- 与制造商无关，几乎所有的硬件和软件制造商执行 OPC 接口标准。
- 多种不同的软件和硬件可以组合在一起。
- 不同制造商的不同设备之间可以交换数据。
- 所有不同的设备可以使用相同的方式编程。
- 很容易通过 C ++ 、Visual Basic、VBA 设计自己的应用程序。
- 可以在网络上使用（通过 DCOM 技术）。

### 9.1.2　OPC 基金会

OPC 基金会是工业自动化领域中主要公司的联盟，包括一些世界上著名的自动化系统和硬件、软件公司，如 Microsoft、Siemens、ABB、Digital、Fisher – Rosemount Systems、Honeywell、SoftwareIntellution Inc 等，主要负责制定 OPC 标准。

目前，OPC 基金会组织已经发布了一系统的 OPC 规范。WinCC 的 OPC 服务器支持下列规范：OPC DA 1.0a 和 OPC DA2.0、OPC HDA 1.1、OPC A&E 1.0、OPC DA XML 1.0 等。

### 9.1.3　OPC 的层次对象模型

OPC 由分处不同层次的三类对象组成：OPC Server 对象、OPC Group 对象和 OPC Item 对象，每类对象都包括一系列接口，如图 9-1 所示。

1. OPC Server 对象

OPC Server 对象提供了一种访问数据的方法，拥有服务器的所有信息，同时也是 Group 对象的容器，主要功能如下：

- 创建和管理 OPC Group 对象。
- 管理服务器内部的状态信息。
- 将服务器的错误代码翻译成描述性语句。
- 浏览 OPC 服务器内部的数据组织结构。

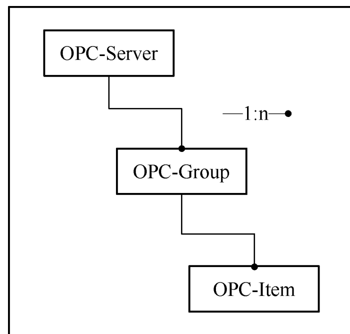


图 9-1　OPC 的层次对象模型

2. OPC Group 对象

OPC Group 对象提供了客户组织数据的一种方法，每个组中都可以定义一个或多个 OPC Item。OPC Group 对象的主要功能表现为：

- 管理 OPC Group 对象的内部状态信息。
- 创建和管理 Items 对象。
- OPC 服务器内部的实时数据存取服务（同步与异步方式）。

Group 是应用程序组织数据的一个单位，客户可对之进行读写，还可设置客户端的数据刷新率。

3. OPC Item 对象

OPC Item 对象是读写数据的最小逻辑单位，一个 Item 与一个具体的过程值相连，每个 Item 虽然代表了与服务器中的某个数据的连接，但它并不是数据源，而仅仅是与数据源的连接。OPC Item 并不提供对外接口，客户不能直接对之进行操作，所有操作都是通过 Group 对象进行的。



客户机操作数据项的一般步骤：

1）通过服务器对象接口枚举服务器端定义的所有数据项。

2）将要操作的数据项加入客户定义的组对象中。

3）通过组对象对数据项进行读写等操作，每个数据项的数据结构包括三个成员变量：即数据值、数据质量和时间戳。

## 9.1.4　OPC 的数据访问方式

OPC 客户端和 OPC 服务器进行数据交互主要有三种方式，即同步方式、异步方式和订阅式数据访问方式。同步方式实现比较简单，当客户端数目较少而且同服务器交换的数据量也比较少的时候可以采用这种方式；异步方式实现相对复杂，需要在客户端程序中实现服务器回调函数。然而当有大量客户和大量数据交换时，采用异步方式能提供高效的性能，从而可以尽量避免阻塞客户数据请求，并最大可能地节省 CPU 和网络资源。订阅式数据访问方式实际上也是异步访问方式的一种。

### 1. 同步访问方式

OPC 服务器将按照 OPC 客户端的要求得到的数据访问结果作为方法的参数返回给 OPC 客户端，OPC 客户端在结果被返回之前必须处于等待状态。

同步访问方式具有以下特点：

- 读取指定 OPC Item 对应的过程数据时，客户端一直要等到读取完为止。
- 写入指定 OPC Item 对应的过程数据时，客户端一直等待写入完成为止。
- 当客户数据较少而且同服务器交互的数据量比较少的时候可以采用这种方式，然而当网络堵塞或大量客户访问时，会造成系统的性能效率下降。
- 编程简单，不需要应答。

### 2. 异步访问方式

OPC 服务器接收到 OPC 客户端的要求后，几乎立即将方法返回。OPC 客户端随后可以进行其他处理。当 OPC 服务器完成数据访问时，OPC 服务器转换角色充当成客户程序，而原来的客户程序此时可以看成是服务器。OPC 服务器主动触发 OPC 客户端的异步访问完成事件，将数据访问结果传送给 OPC 客户端。OPC 客户端在其事件处理程序中接收从 OPC 服务器传来的数据。

异步访问方式具有以下特点：

- 读取指定 OPC Item 对应的过程数据，客户端发出读取要求后立即返回，读取完成时发生读取完成事件，OPC 客户端被调用。
- 写入指定 OPC Item 对应的过程数据，客户端发出写入要求后立即返回，写入完成时发出写入完成事件，OPC 客户端被调用。
- 异步方式的效率更高，能够避免多客户大数据请求的阻塞，并可以最大限度地节省 CPU 和网络资源。

### 3. 订阅式数据访问方式

订阅是数据访问方式并不需要 OPC 客户端向 OPC 服务器提出要求，而是服务器周期性地扫描缓冲区的数据，如果发现数据变化超过一定的幅度时，则更新数据缓冲器，并自动通知 OPC 客户端。这样 OPC 客户端就可以自动接到 OPC 服务器送来的变化通知。

订阅式数据访问方式实际上也属于异步读取方式的一种。采用订阅式数据访问方式的服务器按一定的更新周期（UpdateRate）更新数据缓冲器的数值，如果发现数据有变化时，就会以数据变化事件（DataChange）通知 OPC 客户端。OPC 服务器支持不敏感带（DeadBand），即当 OPC Item 的数据类型是模拟量时，只有当前值与前次值的差的绝对值超过一定的限度时，才更新缓冲器数据并通知 OPC 客户端。由此可以无视模拟值的微小变化，从而减轻 OPC 服务器和 OPC 客户端的负荷。

### 9.1.5　OPC 服务器的自定义和自动化接口

OPC 规范提供了两套接口方案，即自定义接口和自动化接口。开发 OPC 客户端实际上就是对这两套接口的声明和调用。

- 自定义接口（Custom Interface）：是一个 OPC 服务器所必须实现的接口，它描述了 OPC 组件对象的接口和其中的方法，适合熟悉 C++ 语言的客户进行开发。该接口效率高，通过该接口，客户能够发挥 OPC 服务器的最佳性能。
- 自动化接口（Automation Interface）：是可选接口，自动化接口封装了 COM 底层的许多实现，使开发者能比较轻松地实现数据访问，但牺牲了程序的运行速度，适合熟悉 VB 和 VBS 等开发工具的用户。

## 9.2　OPC 在 WinCC 中的应用

西门子是 OPC 基金会组织的成员，因此作为西门子的组态软件，WinCC 全面支持 OPC 协议。WinCC 既可用做 OPC 服务器，又可充当 OPC 客户机。

### 9.2.1　WinCC 支持的 OPC 规范

WinCC 的 OPC 服务器支持下列规范：

- MOPC 数据访问（OPC DA）1.0a、2.0 和 3.0。
- OPC XML 数据访问（OPC XML DA）1.0。
- OPC 历史数据访问（OPC HDA）1.1。
- OPC 报警和事件（OPC A&E）1.02。

#### 1. OPC 数据访问（OPC DA）

OPC 数据访问（OPC DA）是管理实时数据的规范。WinCC V6.0 的 WinCC OPC DA 服务器符合 OPC DA 规范 1.0a 、2.0 和 3.0。OPC DA 是 OPC 中最重要的一个部分，也是用得最多的一种规范。对于一般的用户来说，OPC DA 已经能实现大部分的数据交换要求。

#### 2. OPC XML 数据访问（OPC XML DA）

OPC XML DA 是通过 Internet，使用与平台无关的协议进行通信的一种标准。客户机不再局限于 Windows 环境（DCOM）。其他操作系统，例如 Linux，可以使用 HTTP 协议和 SOAP 接口通过 Internet 监视和交换 OPC 数据。

通过 OPC XML DA 进行数据访问具有与 OPC DA 进行数据访问相似的功能范围。没有为 OPC XML DA 设计与数据改变有关的反馈消息，因为它涉及数据修改，例如对 DCOM 接口的修改。



3. OPC 历史数据访问（OPC HDA）

OPC 历史数据访问（OPC HDA）是访问归档数据的规范。该规范是 OPC DA 规范的扩充。WinCC V 6. x 的 WinCC OPC HDA 服务器符合 OPC HDA 规范 1. 1。

4. OPC 报警和事件（OPC A&E）

OPC 报警和事件（OPC A&E）是访问过程报警和事件的规范。WinCC V 6. x 及以上版本的 WinCC OPC A&E 服务器符合 OPC A&E 规范 1. 02。

注意：在 WinCC 的安装期间，可选择下列 WinCC OPC 服务器进行安装，而 OPC 通信驱动程序（OPC DA 客户机）和 OPC 条目管理器是自动安装的。

- WinCC OPC DA 服务器。
- WinCC OPC HDA 服务器。
- WinCC OPC A&E 服务器。
- WinCC OPC XML DA 服务器。

## 9. 2. 2 在 WinCC 中使用 OPC

要使用 WinCC OPC HDA、WinCC – OPC – XML – DA 服务器和 WinCC OPC A&E 服务器，必须购买许可证。WinCC 选件 WinCC/Connectivity Pack（连接组件包）的许可证必须安装在用做 WinCC OPC HDA 服务器、WinCC OPC XML DA 服务器或 WinCC OPC A&E 服务器的 WinCC 服务器上。

1. WinCC 作为 OPC DA 服务器

WinCC OPC DA 服务器为其他应用程序提供了 WinCC 项目的数据。应用程序能够在同一计算机上运行或在已联网的计算机上运行。例如，以这种方法，WinCC 变量的值可导出到 Microsoft Excel 中，如图 9-2 所示。

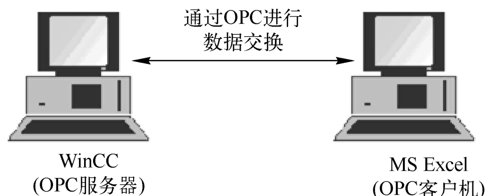



图 9-2 WinCC 作为 OPC DA 服务器

2. WinCC 作为 OPC – XML – DA 服务器

在分布式系统中，WinCC 客户机具有多个 WinCC 服务器的视图。WinCC OPC XML DA 服务器以 Web 页面形式为 OPC XML 客户机提供 OPC 过程数据。可以使用 HTTP 通过 Internet 访问 Web 页面，OPC XML 客户机不再限于局域网。这样，OPC XML 客户机便可通过任何平台以及通过 Intranet 或 Internet 访问 WinCC 运行系统的数据，如图 9-3 所示。

3. 冗余系统中的 WinCC OPC 服务器

在冗余系统中，WinCC 服务器在运行系统中相互监视以便尽早识别服务器故障。使用 OPC 软件接口，WinCC OPC 服务器允许 OPC 客户机访问 WinCC 运行系统数据。

任何基于相应的 OPC 规范的软件程序都可用做 OPC 客户机。因此可以使用 OPC 客户机来集中地监控各种冗余系统。通过创建专有 OPC 客户机，可以满足大多数用户特定需求。如图 9-4 所示。

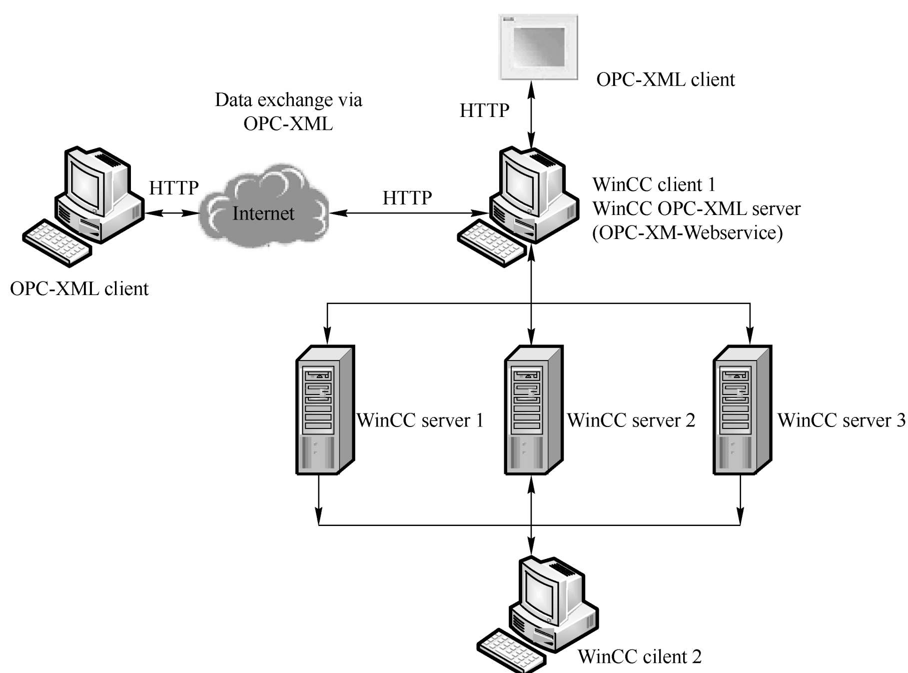



图 9-3　WinCC 作为 OPC－XML－DA 服务器

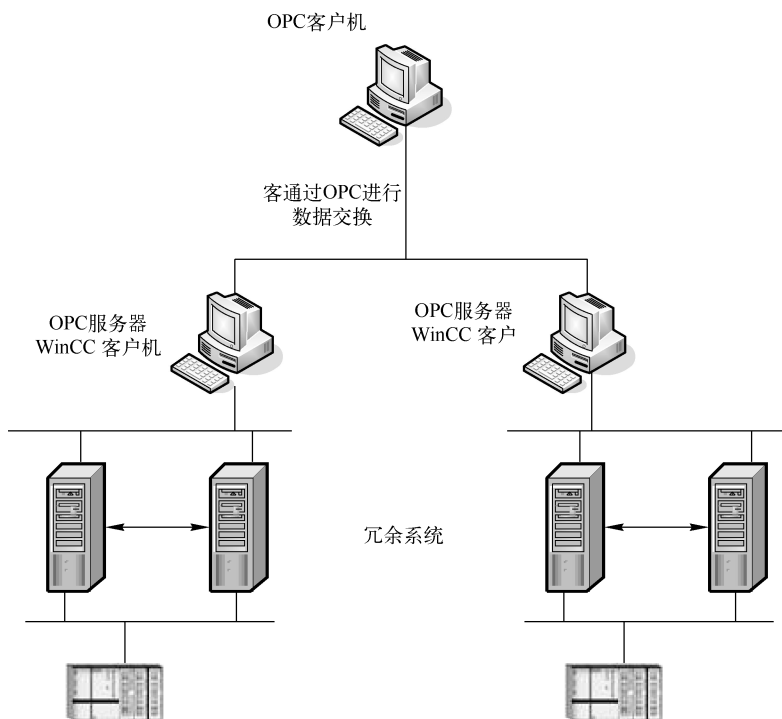



图 9-4　冗余系统中的 WinCC OPC 服务器



## 9.3　WinCC 基于 OPC 的数据访问应用实例

本节通过几个实例介绍 WinCC 在基于 OPC 的数据访问方面的应用，主要是基于 OPC DA 规范方面的应用。

### 9.3.1　WinCC 到 WinCC 的连接

一个 WinCC 作为 OPC 客户端，另一个 WinCC 作为 OPC 服务器，WinCC 所在的两台计算机位于相同的网络中。

当建立 WinCC 到 WinCC 的连接时，WinCC OPC DA 服务器和客户端之间的数据交换通过变量“OPC_Server_Tag”进行。客户端的 WinCC 变量“Client_OPC_Server_Tag_xyz”读取服务器上的 WinCC 变量“OPC_Server_Tag”。如果 WinCC OPC 服务器上的“OPC_Server_Tag”变量的数值改变，WinCC OPC DA 客户端上的 WinCC 变量“Client_OPC_Server_Tag_xyz”的值也改变。客户机上的改变也会在服务器上反映出来。变量值的变化在各自的 I/O 域中显示，如图 9-5 所示。

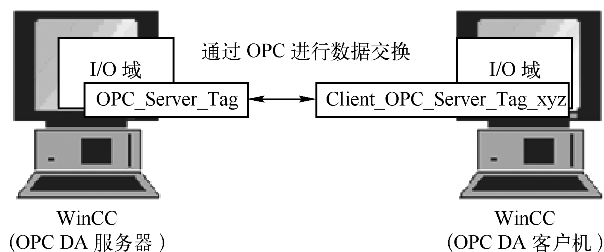



图 9-5　WinCC 到 WinCC 的 OPC 连接

**1. 组态的前提条件**

- 两台带 WinCC 项目的计算机。
- 这两台计算机必须能够通过其 IP 地址被访问。

**2. 组态方法和过程**

为了实现 WinCC 到 WinCC 的 OPC 连接，需要分别组态 WinCC OPC 服务器和 WinCC OPC 客户端。

（1）组态 WinCC OPC 服务器

在服务器 WinCC 项目管理器的变量管理中，创建一个数据类型为“有符号 16 位数”的内部变量“OPC_Server_Tag”。在画面中插入一个对象选项板的“标准”→“智能对象”→“输入/输出域”对象，并在弹出的“I/O 域组态”对话框中，建立与变量“OPC_Server_Tag”的变量连接。

（2）组态 WinCC OPC 客户端

组态 WinCC OPC 客户端的步骤如下：

1）用鼠标右键单击 WinCC 项目管理器浏览窗口的“变量管理”，在弹出的菜单中选择“添加新的驱动程序…”，打开“添加新的驱动程序对话框”，在对话框中选择“OPC.chn”，然后单击“打开”按钮，“变量管理”树形结构中将新增“OPC”一项。

2）在项目管理器中用鼠标右键单击其通道单元“OPC Groups（OPCHN Unit #1）”，在弹出的快捷菜单中选择“系统参数”，打开 OPC 条目管理器，如图 9-6 所示。




图 9-6　OPC 条目管理器

3）选择要用做 OPC DA 服务器的计算机名称，例如图 9-6 中的“DavidLucy”，从其列表中选择“OPCServer. WinCC”，单击“浏览服务器”按钮（OPC DA 服务器上的 WinCC 项目必须处于运行状态），将打开“过滤标准”对话框。

4）在“过滤标准”对话框中，单击“下一步 ->”按钮。在“OPCServer. WinCC...”对话框中选择“OPC_Server_Tag”变量，单击“添加条目”按钮，如图 9-7 所示。

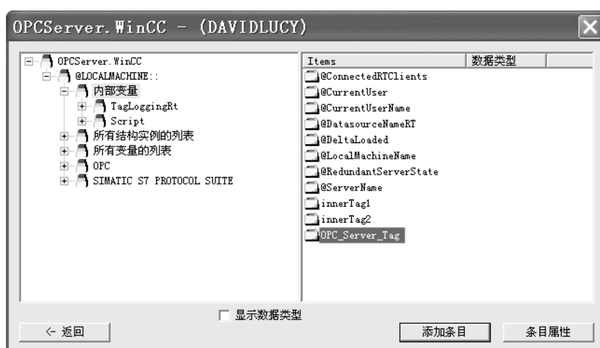



图 9-7　选择访问 OPC DA 服务器的变量

5）如果到 OPC DA 服务器的连接已经存在，继续步骤 6）。如果尚未组态连接，则会显示相应的消息。在显示消息的对话框中单击“是”，将打开“新连接”对话框。输入“OPCServer_WinCC”作为连接的名称，单击“确定”按钮，显示“添加变量”对话框。

6）如图 9-8 所示，在“添加变量”对话框的前缀域中输入“Client_”，在后缀域中输入“_xyz”，选择连接“OPCServer_WinCC”，单击“完成”按钮。

7）在“OPCServer. WinCC ...”对话框中，单击“<-返回”按钮。在“OPC 条目管理器”中，单击“退出”按钮，关闭 OPC 条目管理器。

8）启动 WinCC“图形编辑器”，打开一个新的画面，从对象选项板的“标准”→“智能对象”列表中选择“输入/输出域”对象添加到画面。将打开“I/O




图 9-8　“添加变量”对话框



域组态”对话框。

9）在对话框的“变量”域中输入名称“Client_OPC_Server_Tag_xyz”。将更新周期设置为“有变化时”，将域类型设置为“I/O 域”。关闭对话框并保存画面，单击图形编辑器中的“激活”按钮，以运行 WinCC 项目。

10）在 WinCC OPC DA 服务器和客户端上的 I/O 域中，将显示组态变量的数值。在 WinCC OPC DA 服务器的 I/O 域中输入新的数值。新数值将显示在 WinCC OPC DA 客户端上的 I/O 域内。

### 9.3.2 WinCC 到 Microsoft Excel 的连接

本实例将在 Microsoft Excel 中使用 Visual Basic 编辑器创建一个 OPC DA 客户机。OPC DA 客户机读取 WinCC OPC DA 服务器的 WinCC 项目中的 WinCC 变量，并将值写入单元格中。如果在单元格中输入一个新值，该值将会被传送到 WinCC OPC DA 服务器。

本实例需要在同时安装了 WinCC 和 Microsoft Excel 的计算机上来设置连接，WinCC 与 Microsoft Excel 基于 OPC 的连接示意图如图 9-9 所示。

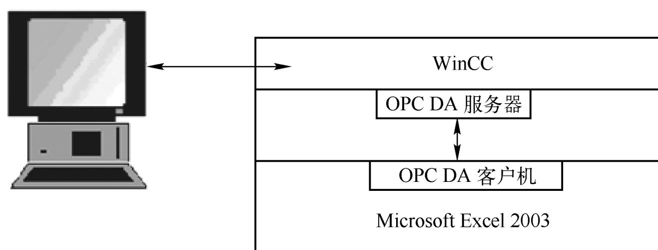



图 9-9　WinCC 与 Microsoft Excel 基于 OPC 的连接

组态 WinCC 到 Microsoft Excel 的连接主要分为两个大的步骤，即：

1）在 Microsoft Excel 的 Visual Basic 编辑器中创建 OPC DA 客户机。

2）组态在 Microsoft Excel 中访问 WinCC 变量。

#### 1. 在 Microsoft Excel 中创建 OPC DA 客户机

在 Microsoft Excel 2003 中创建 OPC DA 客户机的方法和步骤如下：

1）启动一个 Excel 的新工作簿，在 Excel 的“工具”菜单中，单击“宏”→“Visual Basic 编辑器”，打开 Microsoft Excel 的 Visual Basic 编辑器。

2）在 Visual Basic 编辑器中，单击菜单“工具”→“引用（R）…”，打开“引用 - VBAProject”对话框，如图 9 - 10 所示。在“可使用的引用”列表中，勾选“Siemens OPC DAAutomation 2.0”条目，单击“确定”按钮，退出对话框。

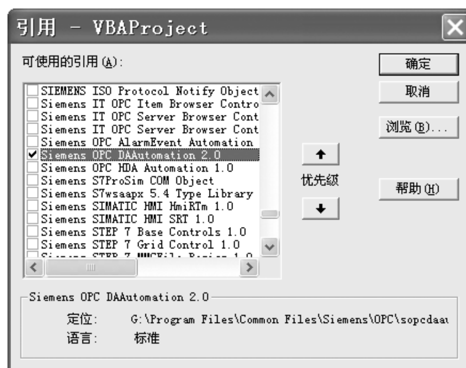



图 9-10　“引用 - VBAProject”对话框

3）双击 Visual Basic 编辑器项目窗口中的“Sheet1”，打开新的代码窗口，在代码窗口中输入以下代码：

```
Option Explicit
Option Base 1
Const ServerName = "OPCServer. WinCC"
Dim WithEvents MyOPCServer As OPCServer
Dim WithEvents MyOPCGroup As OPCGroup
Dim MyOPCGroupColl As OPCGroups
Dim MyOPCItemColl As OPCItems
Dim MyOPCItems As OPCItems
Dim MyOPCItem As OPCItem
Dim ClientHandles(6) As Long
Dim ServerHandles() As Long
Dim Values(6) As Variant
Dim Errors() As Long
Dim ItemIDs(6) As String
Dim GroupName As String
Dim NodeName As String
Dim CellName_T(6) As String
Dim CellName_R(6) As String
Dim CellName_Q(6) As String
Dim CellName_S(6) As String
Dim i As Integer
'————————————————————————————————————————
' 过程 StartClient( ):连接到 OPC 服务器,创建 OPC Group 对象,添加 OPC Item。
'————————————————————————————————————————
Sub StartClient()
  For i = 1 To 6
      ClientHandles(i) = i
   Next i
   GroupName = "MyGroup"
   NodeName = Range("C2"). Value
   For i = 1 To 6
      CellName_T(i) = "B" + CStr(i + 3)
      ItemIDs(i) = Range(CellName_T(i)). Value
   Next i
   Set MyOPCServer = New OPCServer
   MyOPCServer. Connect ServerName, NodeName
   Set MyOPCGroupColl = MyOPCServer. OPCGroups
   MyOPCGroupColl. DefaultGroupIsActive = True
   Set MyOPCGroup = MyOPCGroupColl. Add(GroupName)
```



```
    Set MyOPCItemColl = MyOPCGroup. OPCItems
    MyOPCItemColl. AddItems 6, ItemIDs(), ClientHandles(), ServerHandles(), Errors
    MyOPCGroup. IsSubscribed = True
    Exit Sub
    ErrorHandler:  MsgBox "Error:" & Err. Description, vbCritical, "ERROR"
End Sub
'
' 过程 StopClient():释放对象,断开与 OPC 服务器的连接。
'
Sub StopClient()
    MyOPCGroupColl. RemoveAll
    MyOPCServer. Disconnect
    Set MyOPCItemColl = Nothing
    Set MyOPCGroup = Nothing
    Set MyOPCGroupColl = Nothing
    Set MyOPCServer = Nothing
End Sub
'
' 过程 MyOPCGroup_DataChange():OPC Group 中的值、质量和时间戳发生变化时触发该事件。
'
Private Sub MyOPCGroup_DataChange(ByVal TransactionID As Long, ByVal NumItems As Long,
                        ClientHandles() As Long, itemvalues() As Variant, Qualities() As Long,
                        TimeStamps() As Date)
    For i = 1 To NumItems
    CellName_R(i) = "C" + CStr(ClientHandles(i) + 3)
    Range(CellName_R(i)). Value = itemvalues(i)
    CellName_Q(i) = "D" + CStr(ClientHandles(i) + 3)
    Range(CellName_Q(i)). Value = Hex(Qualities(i))
    CellName_S(i) = "E" + CStr(ClientHandles(i) + 3)
    Range(CellName_S(i)). Value = CStr(TimeStamps(i))
    Next i
End Sub
```

4）返回 Excel，根据图 9-11 所示编辑 Excel 的 Sheet1。其中标识为“写变量值到 WinCC”的对象是一个按钮，添加按钮的方法：单击 Excel 菜单“视图”→“工具栏”→“Visual Basic”，将打开 Visual Basic 工具栏，单击该工具栏的“控件工具箱”按钮，激活“控件工具箱”，在“控件工具箱”中选中“命令按钮”，鼠标变成十字形，在 Excel 的适当位置按鼠标左键绘制按钮并拖放到合适大小，如图 9-11 所示。用鼠标右键单击添加的按钮，在弹出的菜单中选择“属性”，打开“属性”对话框，如图 9-12 所示，在属性列表中找到“Caption”属性，在其后的文本框中输入“写变量值到 WinCC”。

5）双击添加的按钮，将进入 Visual Basic 编辑器，并自动创建过程 CommandButton1_Click()，在该过程中输入以下代码：




图 9-11　编辑 Excel 的 Sheet1

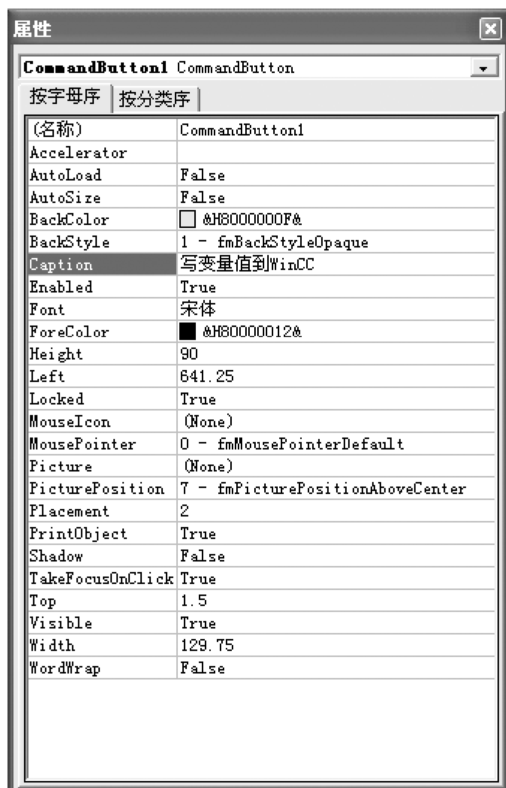



图 9-12　按钮对象的“属性”窗口

'

' 过程 CommandButton1_Click( ):按钮的单击事件,在 Excel 运行宏的状态时单击该按钮,将单元格 G4 ~ G9 的值写入与其对应的 WinCC 变量中

'



```
Private Sub CommandButton1_Click( )
    For i = 1 To 6
      CellName_R(i) = "G" + CStr(i + 3)
      Values(i) = Range(CellName_R(i)). Value
    Next i
  MyOPCGroup. SyncWrite 6, ServerHandles, Values, Errors
End Sub
```

6）在 Visual Basic 编辑器的“文件”菜单中选择“保存”，从“文件”菜单中选择“关闭”，退出 Visual Basic 编辑器并返回到 Microsoft Excel。

7）在 Excel 的“文件”菜单中选择“保存”，将该 Excel 文档保存为“WinCC_OPC_Excel. xls”。

### 2. 组态在 Microsoft Excel 中访问 WinCC 变量

1）WinCC OPC DA 服务器的 WinCC 项目中，组态 6 个名称分别为“OPC_Excel_1” ~ “OPC_Excel_6”且数据类型为“有符号 16 位数”的内部变量。在 WinCC 图形编辑器中新建一个画面，然后在画面中添加 6 个“输入/输出域”，并分别与组态的 6 个内部变量建立变量连接。激活该 WinCC 项目。

2）打开 Excel 文档“WinCC_OPC_Excel. xls”，在弹出的“安全警告”框中选择“启用宏”。在 Sheet 的单元格 C2 中，输入用做 OPC 服务器的计算机名称。

3）在 Excel 的“工具”菜单中，选择“宏”→“宏（M）…”命令，“宏”对话框将打开。从宏列表中选择条目“Sheet1. StartClient”，单击“运行”以启动 OPC 客户机。WinCC 变量“OPC_Excel_1” ~ “OPC_Excel_6”的值会写入单元格 C4 ~ C9 中，质量代码写入 D4 ~ D9 中，时间戳写入 E4 ~ E9 中，如图 9-13 所示。

Microsoft Excel - WinCC_OPC_Excel

文件(F) 编辑(E) 视图(V) 插入(I) 格式(O) 工具(T) 数据(D) 窗口(W) WinCC 帮助(H) Adobe PDF

OPC-WinCC

写变量值到WinCC

| | | | | | |
|---|---|---|---|---|---|
| OPC Server的计算机名 | | | | | |
| WinCC变量名 | 实际值（RealValue） | 质量（Quality） | 时间戳（TimeStamp） | | |
| OPC_Excel_1 | 700 | C0 | 2012-10-1 16:48 | | 33 |
| OPC_Excel_2 | 143 | C0 | 2012-10-1 16:48 | | 34 |
| OPC_Excel_3 | 210 | C0 | 2012-10-1 16:48 | | 35 |
| OPC_Excel_4 | 400 | C0 | 2012-10-1 16:48 | | 36 |
| OPC_Excel_5 | 56 | C0 | 2012-10-1 16:48 | | 37 |
| OPC_Excel_6 | 77 | C0 | 2012-10-1 16:48 | | 38 |

图 9-13　运行结果

4）在单元格 G4 ~ G9 中输入新的数值，单击“写变量值到 WinCC”按钮，更改后的数值将显示在 WinCC OPC 服务器的过程画面中对应的 I/O 域内。

5）在 Excel 的“工具”菜单中，选择“宏”→“宏（M）…”命令，“宏”对话框将打开，从宏列表中选择条目“Sheet1. StopClient”，单击“运行”按钮，将停止 OPC 客户机。

### 9.3.3　WinCC 与 SIMATIC NET 的 OPC DA 连接

在安装 SIMATIC NET 期间，可以选择要安装的 OPC 服务器。在本实例中，组态 WinCC 至 SIMATIC NET S7 OPC 服务器的连接，WinCC 作为 OPC DA 的客户机，通过 SIMATIC NET S7 OPC 服务器访问自动化设备的数据，如图 9-14 所示。过程变量的当前值显示在 WinCC OPC 客户机的一个 I/O 域中。一旦 SIMATIC NET S7 OPC 服务器上的变量值改变，更改后的数值将显示在过程画面上。反之，在 I/O 域中输入的数值会发送到自动化设备上。

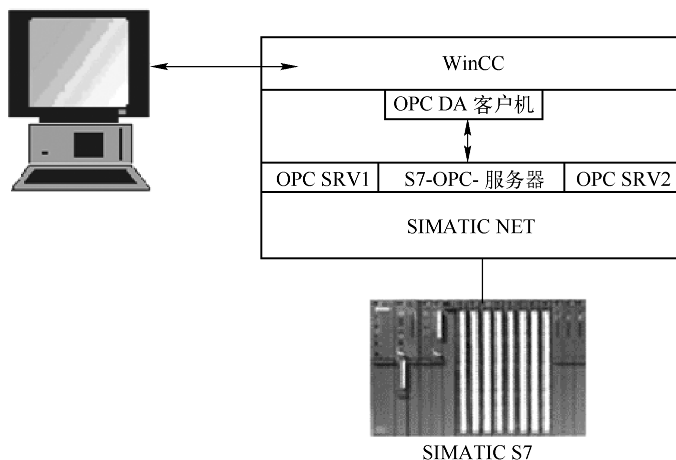



图 9-14　WinCC 与 SIMATIC NET 的 OPC DA 连接

要组态 WinCC 与 SIMATIC NET 之间的 OPC DA 连接，必须在计算机上同时安装有 WinCC 和 SIMATIC NET 软件，且已经组态 SIMATIC NET S7 OPC 服务器。

为了组态 WinCC 与 SIMATIC NET 的 OPC DA 连接，需要完成以下组态：

- 添加变量到 SIMATIC NET S7 OPC 服务器。
- 在 WinCC 组态对 SIMATIC NET S7 OPC 服务器的变量的访问。

**1. 添加变量到 SIMATIC NET S7 OPC 服务器**

在添加变量到 SIMIC NET S7 OPC 服务器之前，必须在 SIMATIC NET 或 SIMATIC Manager 中组态 SIMATIC NET S7 OPC 服务器与自动化设备（例如 S7 -300/400 PLC，但 CPU 必须支持 OPC 访问）的 S7 连接。图 9-15 所示是在 SIMATIC Manager 中组态的 SIMATIC NET S7 OPC 服务器与 S7 -400 PLC 的 CPU 412 -2 DP 之间的 S7 连接。组态方法请参考 SIMATIC STEP7 或 SIMATIC NET 的相关文档。






图 9-15　SIMATIC NET S7 OPC 服务器与 S7-400 PLC 的 S7 连接

添加变量到 SIMIC NET S7 OPC 服务器的方法和步骤如下：

1）通过单击菜单“开始”→“程序”→“SimaticNet”→“OPCServer”→“OPC-Scout”，打开“OPC Scout”，如图 9-16 所示。

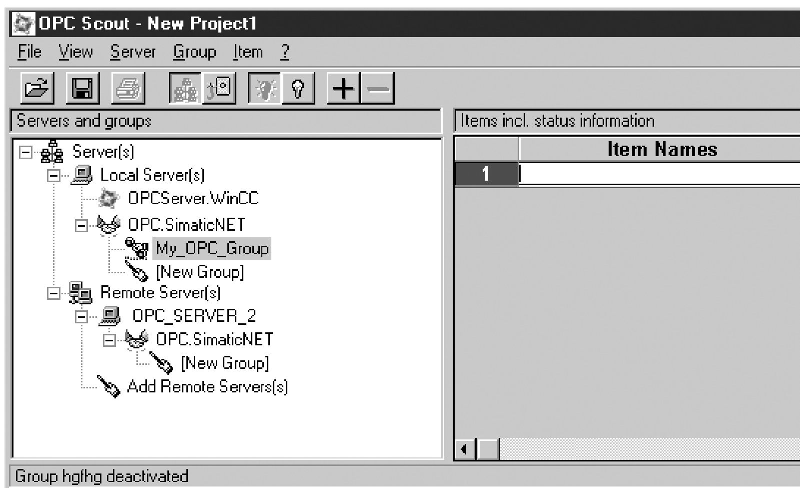



图 9-16　OPC Scout

2）选择“Local Server(s)”（本地服务器）下的“OPC. SimaticNet”。如果 SIMATIC S7 OPC 服务器不是运行在同一台计算机上，则选择“Server(s)”（服务器）快捷菜单中的“Add Remote Server(s)”（添加远程服务器）。在“Add Remote Server(s)”（添加远程服务器）对话框中输入用做 OPC 服务器的计算机名称，然后单击“OK”按钮关闭对话框。

3）用鼠标右键单击“OPC. SimaticNet”，在弹出的菜单中选择“Connect”（连接）。

显示“Add Group”（添加组）对话框，为该组输入一个名称，例如“My_OPC_Group”，单击“OK”按钮关闭对话框。

4）用鼠标右键单击所添加的组，在弹出的菜单中选择“Add Item”（添加条目），打开“OPC Navigator”（OPC 浏览器），如图 9-17 所示。

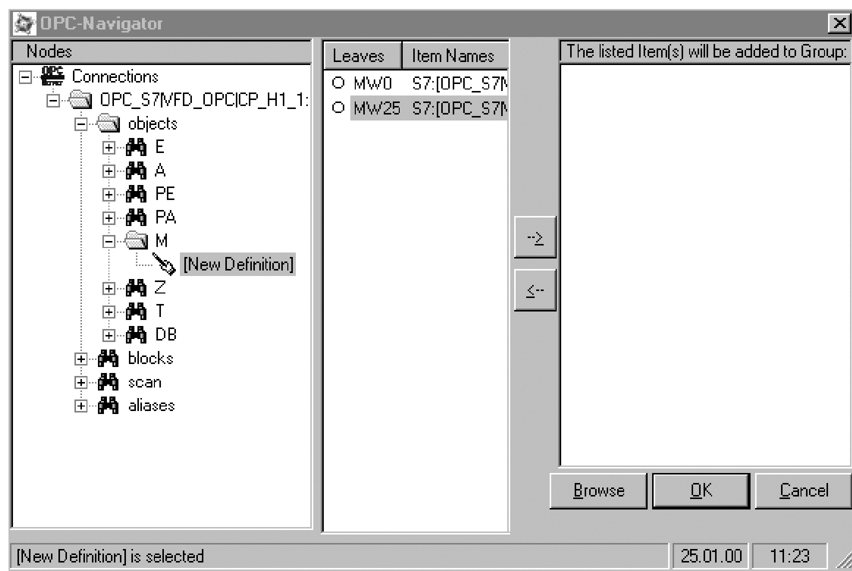



图 9-17　OPC Navigator

5）在“OPC Navigator”中选择“Objects”（对象）下的“M”（标记）。双击“[New Definition]”（新的定义），打开“Define New Tag”（定义新的变量）对话框，如图 9-18 所示。在对话框中输入相应的参数，单击“OK”按钮关闭对话框。

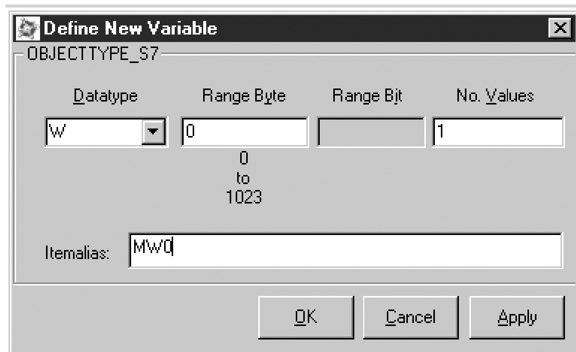



图 9-18　“定义新的变量”对话框

6）在 OPC 浏览器的“Leaves”区域中标记变量“MW0”，单击“-->”按钮将其添加到右边的列表中。单击 OPC 浏览器的“OK”按钮退出 OPC 浏览器。

**2. 在 WinCC 组态对 SIMATIC NET S7 OPC 服务器的变量的访问**

这里的组态任务是在 WinCC OPC DA 客户机的 WinCC 项目中组态 WinCC 变量，该变量将访问 SIMATIC NET S7 OPC 服务器地址空间中的变量“MW0”，变量的值显示在 I/O



域中，组态方法和步骤如下：

1）用鼠标右键单击 WinCC 项目管理器浏览窗口的“变量管理”，在弹出的菜单中选择“添加新的驱动程序…”，打开“添加新的驱动程序对话框”，在对话框中选择“OPC. chn”，然后单击“打开”按钮，“变量管理”树形结构中将新增“OPC”一项。

2）用鼠标右键单击“OPC Groups（OPCHN Unit#1）”快捷菜单中的“系统参数”，打开 OPC 条目管理器。

3）选择用做 OPC DA 服务器的计算机名称，如果是本地计算机可选择“\\ <LOCAL>”，从该计算机的列表中选择“OPC. SIMATICNet”。单击“浏览服务器”按钮，将打开“过滤标准”对话框。

4）在“过滤标准”对话框中，单击“下一步 - >”按钮，“OPC. SIMATICNet. .”对话框将打开。选择“MW0”变量，单击“添加条目”按钮。

5）如果到 SIMATIC NET OPC 服务器的连接已经存在，继续步骤 6），如果尚未组态连接，则会显示相应的消息。在显示消息的对话框中单击“是”，打开“新连接”对话框，在文本框中输入“OPC_SIimaticNET”作为连接的名称，单击“OK”按钮，打开“添加变量”对话框。

6）在“添加变量”对话框的前缀域中输入“Client_”，在后缀域中输入“_xyz”。选择连接“OPC_SimaticNET”，单击“完成”按钮。

7）在“OPC. SIMATICNet. .”对话框中，单击“< - 返回”按钮，在“OPC 条目管理器”中，单击“退出”按钮关闭 OPC 条目管理器。

8）启动图形编辑器并打开画面，从对象选项板的“标准”→“智能对象”列表中选择“输入/输出域”对象添加到画面中，将打开“I/O 域组态”对话框。

9）在对话框的“变量”域中，输入名称“Client_MW0_xyz”，将更新周期设置为“有变化时”，将域类型设置为“I/O 域”。

10）关闭对话框并保存画面，单击图形编辑器中的“激活”按钮，运行 WinCC 项目。

11）S7 变量的当前值显示在 WinCC OPC DA 客户机的 I/O 域中，数值发生变化时将立即更新显示。在 I/O 域中输入数值，更改后的数值将传递到自动化设备。

### 9.3.4 WinCC 与 S7 - 200 PLC 的 OPC DA 连接

S7 - 200 系列 PLC 是西门子公司的自动化产品，但是 WinCC 却不能直接监控 S7 - 200 系列 PLC 组成的控制系统。这是因为 S7 - 200 系列 PLC 常用的通信协议是 PPI 协议，而 WinCC 软件中并没有集成 PPI 协议。为此，西门子公司专门推出了用于 S7 - 200 PLC 的 OPC 服务器软件 PC Access，这样 WinCC 可以通过 OPC 接口访问 S7 - 200 PLC 的数据。在这种访问方式中，WinCC 作为 OPC DA 客户机，而 PC Access 的角色是 OPC DA 服务器，如图 9-19 所示。

PC Access 软件可以与任何标准的 OPC 客户机进行通信，为 OPC 客户机提供数据信息，它自带 OPC 客户测试端，用户可以方便地检测其项目的通信及配置的正确性。

组态 WinCC 与 S7 - 200 PLC 的 OPC DA 连接，需要完成以下组态：

- 应用 PC Access 组态访问 S7 - 200 PLC 的 OPC DA 服务器。
- 在 WinCC 中组态对 OPC DA 服务器的变量的访问。

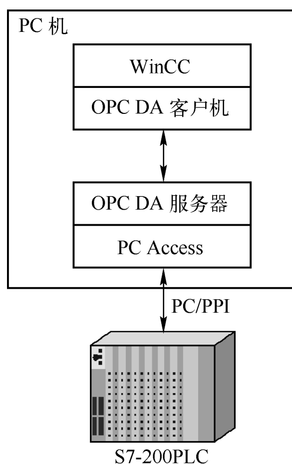



图 9-19　WinCC 与 S7－200 PLC 的 OPC DA 连接

### 1. 应用 PC Access 组态访问 S7－200 PLC 的 OPC 服务器

应用 PC Access 组态访问 S7－200 PLC 的 OPC 服务器的方法和步骤如下：

1）双击 Windows 桌面上的 PC Access 软件图标，打开 S7－200 PC Access 组态窗口。在工具栏中单击“New”按钮，将在左边的窗口中新增“MicroWin（COM1）”一项。

2）用鼠标右键单击“MicroWin（COM1）”，在弹出的菜单中选择“PG/PC Interface…”命令，打开“设置 PG/PC 接口”对话框设置通信方式，如图 9-20 所示，选择“PC/PPI cable（PPI）”通信方式。

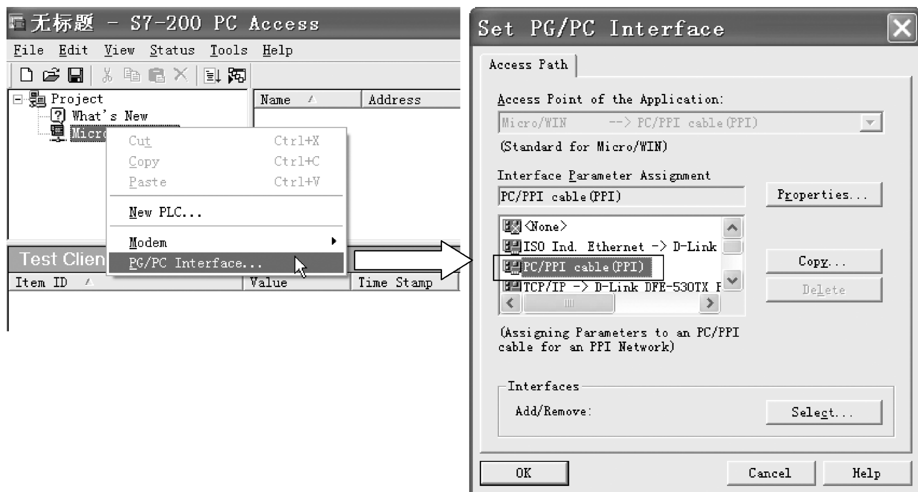



图 9-20　设置 PC Access 的通信通道

3）用鼠标右键单击“MicroWin（COM1）”，在弹出的菜单中选择“New PLC…”命令，添加一个新的 S7－200 PLC 站（最多可添加 8 个 S7－200 PLC），并在弹出的“PLC Properties”对话框中定义 PLC 的名称（Name）“S7－200_OPC_WinCC”，设置 CPU 的网络地址（Network Address），该地址应该与 PLC 内的地址一致，如图 9-21 所示。

4）用鼠标右键单击新添加的 PLC 站点“S7－200_OPC_WinCC”，在弹出的菜单中选择“New”→“Item…”，添加 PLC 内存数据的条目并定义内存数据，如图 9-22 所示。



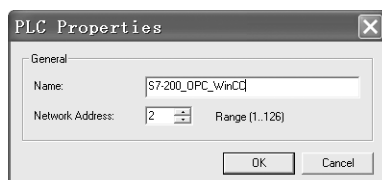



图 9-21　设置 PLC 的属性

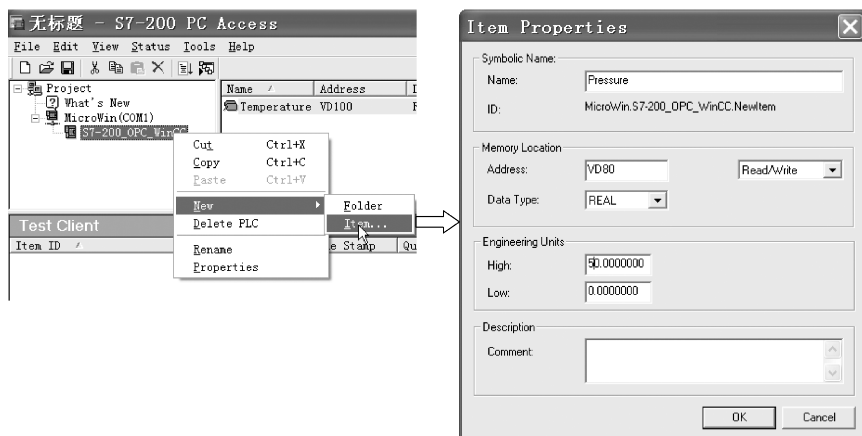



图 9-22　添加“S7 - 200_OPC_WinCC”站的条目

5）选中添加的条目，用鼠标拖放到“Test Client”（测试客户机）窗口中，单击工具栏的“Test Client Status”（测试客户机状态）按钮，测试客户机的状态。如果配置及通信正确，将在“Value”列显示条目的数据值，并在“Quality”（质量）一列中显示“Good”，否则将显示“Bad”。

注意：在测试客户机状态之前，需要单击工具栏的“保存”按钮对项目进行保存。组态完成后也必须保存整个的配置文件，这样 OPC 客户机才能找到 PC Access OPC Server 的组态，PC Access 软件创建的项目文件的扩展名为 . pca。

**2. 在 WinCC 中组态对 OPC DA 服务器的变量的访问**

在本实例中，WinCC 是作为 OPC DA 客户机的，需要在 WinCC 中组态对 S7 - 200 OPC Server 的变量的访问，组态方法和步骤如下：

1）用鼠标右键单击 WinCC 项目管理器浏览窗口的“变量管理”，在弹出的菜单中选择“添加新的驱动程序…”，打开“添加新的驱动程序对话框”，在对话框中选择“OPC. chn”，然后单击“打开”按钮，“变量管理”树形结构中将新增“OPC”一项。

2）用鼠标右键单击“OPC Groups（OPCHN Unit#1）”快捷菜单中的“系统参数”，打开 OPC 条目管理器。

3）选择用做 OPC DA 服务器的计算机名称，如果是本地计算机可选择“\\<LOCAL>”，从该计算机的列表中选择“S7200. OPCServer”。单击“浏览服务器”按钮，将打开“过滤标准”对话框，如图 9-23 所示。

4）在“过滤标准”对话框中，单击“下一步 - >”按钮，“S7200. OPCServer . . ”对话框将打开。选择“Pressure”变量，单击“添加条目”按钮，如图 9-24 所示。




图 9-23　OPC 条目管理器




图 9-24　“S7200. OPCServer . . ” 对话框

5）如果到 S7200. OPCSserver 的连接已经存在，继续步骤 6），如果尚未组态连接，则会显示相应的消息。在显示消息的对话框中单击“是”，打开“新连接”对话框，在文本框中输入“S7200_OPCServer”作为连接的名称，如图 9-24 所示，单击“OK”按钮，打开“添加变量”对话框。

6）在“添加变量”对话框的前缀域中输入“S7200”，选择连接“S7200_OPCServer”，单击“完成”按钮，如图 9-25 所示。




图 9-25　添加变量的前缀和后缀

7）在“S7200. OPCServer . . ”对话框中，单击“<- 返回”按钮，在“OPC 条目管理器”中，单击“退出”按钮关闭 OPC 条目管理器。

8）启动图形编辑器并打开画面，从对象选项板的“标准”→“智能对象”列表中选择“输入/输出域”对象添加到画面中，将打开“I/O 域组态”对话框。

9）在对话框的“变量”域中，输入名称“S7200Pressure”，将更新周期设置为“有变化时”，将域类型设置为“I/O 域”。

10）关闭对话框并保存画面，单击图形编辑器中的“激活”按钮，运行 WinCC 项目。

11）S7 变量的当前值显示在 WinCC OPC DA 客户机的 I/O 域中，数值发生变化时将立即更新显示。在 I/O 域中输入数值，更改后的数值将传递到 S7－200 PLC。

# 附录

## 附录 A　C 运算符

| 运　算　符 | 名　　称 | 举　　例 |
|---|---|---|
| & | 二进制与运算符 | C = A&B 变量 C 等于 A 和 B 中都置位<br>A = 01001110<br>B = 10100111<br>---------------<br>C = 00000110 |
| \| | 二进制或运算符 | C = A \| B 变量 C 等于 A 或者 B 中某一置位即可<br>A = 01001110<br>B = 10100111<br>---------------<br>C = 11101111 |
| ^ | 二进制异或运算符 | C = A^B 变量 C 等于 A 或者 B 中不同的位<br>A = 01001110<br>B = 10100111<br>---------------<br>C = 11101001 |
| ~ | 二进制非运算符 | C = ~A 变量 C 等于 A 状态取反<br>A = 01001110<br>---------------<br>C = 10110001 |
| == | 比较相等<br>例如：<br>if（A == 10）{Then}<br>else {otherwise} | C = A == B 如 A 等于 B，C 为真，否则<br>C = FALSE（0）<br>A = 10　　　　A = 10<br>B = 22　　　　B = 10<br>------------　　------------<br>C = 0　　　　C = TRUE |
| && | 逻辑与运算符<br>例如：<br>if（A&&B）{Then}<br>else {otherwise} | C = A&&B 如 A 和 B 都为非零数则 C 为真，否则 C = FALSE（0）<br>A = 10　　　A = 10　　　A = 0<br>B = 22　　　B = 0　　　B = 99<br>----------　　----------　　----------<br>C = 22　　　C = 0　　　C = 0 |
| \|\| | 逻辑或运算符<br>例如：<br>if（A \|\| B）{Then}<br>else {otherwise} | C = A \|\| BA 或 B 有一个为非零数则 C 为真，否则 C = FALSE（0）<br>A = 10　　　A = 0　　　A = 0<br>B = 22　　　B = 33　　　B = 0<br>----------　　----------　　----------<br>C = 1　　　C = 33　　　C = 0 |
| ! | 逻辑非 | C = A 非，C 等于 A 的取反 |

# 附录 B　VB 运算符

| 运　算　符 | 名　　称 | 举　　例 |
|---|---|---|
| AND | 二进制与运算 | C = A AND BC 等于 A 和 B 的与结果<br>A = 01001110<br>B = 10100111<br>---------------<br>C = 00000110 |
| OR | 二进制或运算 | C = A OR BC 等于 A 和 B 的或结果<br>A = 01001110<br>B = 10100111<br>---------------<br>C = 11101111 |
| XOR | 二进制异与运算 | C = A XOR BC 等于 A 和 B 的异或结果<br>A = 01001110<br>B = 10100111<br>---------------<br>C = 11101001 |
| NOT | 二进制非运算 | C = ~AC 等于 A 的取反状态<br>A = 01001110<br>---------------<br>C = 10110001 |
| = | 比较指令<br>例如：<br>if（A = 10）Then<br>…<br>Else<br>…<br>End If | C = A = BA 和 B 相等则 C 为真，否则 C = FALSE（0）<br>A = 10　　A = 10<br>B = 22　　B = 10<br>------------　------------<br>C = 0　　C = TRUE |
| AND | 逻辑与运算<br>例如：<br>If（A AND B）Then<br>…<br>Else<br>…<br>End If | C = A AND B 如 A 和 B 都为非零数则 C 为真，否则 C = FALSE（0）<br>A = 10　　A = 10　　A = 0<br>B = 22　　B = 0　　B = 99<br>----------　----------　----------<br>C = 22　　C = 0　　C = 0 |
| OR | 逻辑或运算<br>例如：<br>If（A OR B）Then<br>…<br>Else<br>…<br>End If | C = A OR BA 或 B 有一个为非零数则 C 为真，否则 C = FALSE（0）<br>A = 10　　A = 0　　A = 0<br>B = 22　　B = 33　　B = 0<br>----------　----------　----------<br>C = 1　　C = 33　　C = 0 |

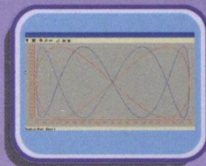
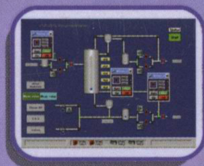
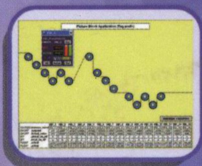
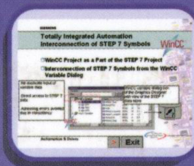
# 参 考 文 献


[1] 刘华波, 王雪, 何文雪, 张赟宁. 组态软件 WinCC 及其应用 [M]. 北京: 机械工业出版社, 2009.
[2] 陈瑞阳. 工业自动化技术 [M]. 北京: 机械工业出版社, 2011.
[3] 甄立东. 西门子 WinCC V7 基础与应用 [M]. 北京: 机械工业出版社, 2011.
[4] 西门子 (中国) 有限公司自动化与驱动集团. 深入浅出西门子 WinCC V6 [M]. 2 版. 北京: 北京航空航天大学出版社, 2005.
[5] 陈瑞阳, 席巍, 宋柏青. 西门子工业自动化项目设计实践 [M]. 北京: 机械工业出版社, 2009.
[6] 梁绵鑫, 罗艳红, 边春元, 渠丰沛等. WinCC 基础及应用开发指南 [M]. 北京: 机械工业出版社, 2009.
[7] 西门子自动化培训中心. WinCC 基础课程. 2006.
[8] 西门子 (中国) 有限公司. WinCC 数据报表实现方法介绍. 2010.
[9] 西门子 (中国) 有限公司. 组态来自于 PLC 的编号消息. 2008.
[10] 西门子 (中国) 有限公司. 如何将 WinCC 归档的历史数据导出到 Excel CSV 文件. 2010.
[11] 西门子 (中国) 有限公司. WinCC 与 Excel、VB 的本地 DDE 通信. 2011.
[12] 西门子 (中国) 有限公司. WinCC 兼容性列表. 2010.
[13] SIEMENS. WinCC 组态手册. 1999.
[14] SIEMENS. WinCC 通信手册. 1999.
[15] SIEMENS. SIMATIC HMI WinCC V6.2 Getting Started. 2007.
[16] SIEMENS. SIMATIC HMI WinCC V6.2 ASIA 手册. 2007.
[17] 西门子 (中国) 工业业务领域工业自动化与驱动技术网址.
http://www.ad.siemens.com.cn/Service/Technical_support.asp.

在线互动交流平台

官方微博：http://weibo.com/cmpjsj
豆 瓣 网：http://site.douban.com/139085/
读者信箱：cmp_itbook@163.com

# WinCC组态技巧与技术问答


## 内容提要

本书按照组态HMI系统的一般过程，系统地介绍了组态软件WinCC在创建HMI项目、组态HMI系统与控制器的通信连接、组态过程监控画面、组态消息系统、组态过程变量记录系统、组态数据报表系统等过程中涉及的基本概念、基本理论以及组态方法和组态技巧，对用户在应用WinCC时遇到的一些技术问题和难题进行了详细的分析并提出了有效的解决方法。

本书可作为高等院校机械工程及自动化、电气工程及自动化、计算机控制等相关专业的教材，也可作为企业现场工程技术人员的培训教材或参考书。


地址：北京市百万庄大街22号
邮政编码：100037
电话服务
社服务中心：010-88361066
销售一部：010-68326294
销售二部：010-88379649
读者购书热线：010-88379203
网络服务
教材网：http://www.cmpedu.com
机工官网：http://www.cmpbook.com
机工官博：http://weibo.com/cmp1952
封面无防伪标均为盗版

上架指导 工业技术/电气工程

ISBN 978-7-111-40929-8

策划编辑◎时 静 / 封面设计◎









定价：59.00元（含1DVD）